



UNIVERSIDAD CARLOS III DE MADRID  
DEPARTAMENTO DE INFORMÁTICA

TESIS DOCTORAL

ALGORITMO INTELIGENTE PARA EL  
RECONOCIMIENTO Y ACOTACIÓN SEMÁNTICA DE  
PRIMITIVAS EN UNA NUBE DE PUNTOS 3D

AUTOR:

YULIANA PÉREZ GALLARDO

DIRECTORES:

DR. ÁNGEL GARCÍA CRESPO  
DR. JOSÉ LUIS LÓPEZ CUADRADO

LEGANÉS, JULIO 2015





UNIVERSIDAD CARLOS III DE MADRID  
DEPARTAMENTO DE INFORMÁTICA

TESIS DOCTORAL

ALGORITMO INTELIGENTE PARA EL RECONOCIMIENTO Y ACOTACIÓN  
SEMÁNTICA DE PRIMITIVAS EN UNA NUBE DE PUNTOS 3D

AUTOR: YULIANA PÉREZ GALLARDO  
DIRECTORES: DR. ÁNGEL GARCÍA CRESPO  
DR. JOSÉ LUIS LÓPEZ CUADRADO

Firma del Tribunal Calificador:

		Firma
Presidente:	_____	_____
Vocal:	_____	_____
Secretario:	_____	_____

Calificación:

Leganés,                      Julio de 2016





## Agradecimientos

*Principalmente mis padres por el amor, dedicación  
y ejemplo, simplemente por ser los mejores.  
A Joel y Jonatan por apoyarme en mis locuras y  
estar ahí siempre para mí, los quiero.  
A Daniel y Cynthia por los consejos, los  
viajes y por aguantarme en mis malos ratos.  
A mis directores José y Ángel por la infinita  
paciencia y consejos durante estos cuatro años.  
Cómo no mencionar a todos los que forman parte  
de mi familia aquí en España José Luis, Gustavo,  
Javier, Mateusz, Alex, Gandhi, Ana, Los Mora,  
por hacer mi estancia tan divertida.  
Finalmente a mis colegas, familia y amigos  
que me han apoyado desde México.  
A todos gracias.*



## Resumen

Describir detalladamente una escena digital, tal y como la percibe un humano no es una tarea fácil, se requiere de muchos factores como son conocimiento previo, capacidad de análisis y reconocimiento. Factores que no es posible encontrarlos en una sola ciencia, de esta manera, la hibridación del uso de maquetas digitales, la ingeniería inversa y tecnologías semánticas puede mejorar el proceso. El objetivo general de la tesis es definir un algoritmo de ingeniería inversa, capaz de reconocer elementos a partir de primitivas geométricas, específicamente cilindros, toros y esferas en una nube de puntos captados vía un escáner de escenas y asociarlos lógicamente, semántica y físicamente con elementos que se pueden encontrar en una escena real. Esto permitirá la generación de una maqueta digital enriquecida en donde los objetos serán descritos con un alto grado de detalle, proporcionando no solo la forma, sino también características geométricas como alto, largo, diámetro; su topología e información semántica de acuerdo al rol que representa cada elemento en la escena. En esta tesis se presenta el algoritmo propuesto, una implementación del mismo en un contexto industrial real y una evaluación que mida la calidad de la clasificación de dicho algoritmo comparándolo como un algoritmo de reconocimiento de formas geométricas tradicional.

**Keywords:** Algoritmo inteligente, reconocimiento, acotación semántica, nube de puntos.

## Abstract

Describing in detail a digital scene, as it is perceived by a human is not an easy task, it requires many factors such as prior knowledge, analytical skills and recognition. These factors can not find a single science, in this sense, hybridization of using digital models, reverse engineering and semantic technologies can improve this process. The thesis overall aim is defining a reverse engineering algorithm, able to recognize elements from geometric primitives, specifically cylinders, tori and spheres in a cloud of points captured via a scanner, and associate them to logic, semantics and physic information to describe a real scene. This will allow the generation of a digital mockup-enriched where objects are described with a high degree of detail, providing not only the shape but also geometric characteristics such as height, length, diameter; its topology and semantic information according to the role that represents each item in the scene. In this thesis an intelligent algorithm, and its implementation in a real industrial context is presented. Also an assessment that measures the quality of the classification of this algorithm, comparing it with a traditional recognition algorithm is introduced.

**Keywords:** Algoritmo inteligente, reconocimiento, acotación semántica, nube de puntos.

<b>Abstract</b>	<b>iii</b>
<b>Resumen</b>	<b>iv</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación de la investigación . . . . .	2
1.2 Justificación . . . . .	3
1.3 Objetivo general y objetivos específicos . . . . .	4
1.3.1 Objetivo General . . . . .	4
1.3.2 Objetivos específicos . . . . .	4
1.4 Organización de la tesis doctoral . . . . .	5
<b>2 Estado del Arte</b>	<b>7</b>
2.1 Reconocimiento de objetos 3D . . . . .	7
2.1.1 Proceso general de reconocimiento de objetos en nube de puntos . .	9
2.1.2 Descriptores Visuales . . . . .	10
2.1.3 Enfoques del reconocimiento de objetos . . . . .	15
2.2 Tecnologías semánticas . . . . .	17
2.2.1 La web semántica . . . . .	17
2.2.2 Conclusiones del estado del arte . . . . .	26
<b>3 Hipótesis a resolver</b>	<b>29</b>
3.1 Planteamiento de las hipótesis a resolver . . . . .	29
3.2 Hipótesis . . . . .	30
3.3 Validación de las hipótesis . . . . .	30

<b>4</b>	<b>Algoritmo inteligente</b>	<b>33</b>
4.1	Descripción general . . . . .	33
4.2	Proceso de reconocimiento de primitivas . . . . .	37
4.2.1	Pre-proceso . . . . .	37
4.2.2	Definición de descriptores . . . . .	43
4.2.3	Clasificación . . . . .	43
4.3	Enriquecimiento semántico . . . . .	44
4.3.1	Definición de la ontología . . . . .	45
4.3.2	Cálculo de características geométricas . . . . .	46
4.3.3	Cálculo de topología . . . . .	46
4.3.4	Validación Semántica . . . . .	50
4.3.5	Validación por expertos . . . . .	51
4.4	Conclusiones del Algoritmo inteligente . . . . .	51
<b>5</b>	<b>Implementación: Caso de uso contexto industrial</b>	<b>53</b>
5.1	Datos fuente . . . . .	53
5.1.1	Nubes de puntos 3D . . . . .	54
5.2	Implementación del algoritmo propuesto . . . . .	57
5.2.1	Proceso de reconocimiento de primitivas . . . . .	57
5.2.2	Proceso de enriquecimiento semántico . . . . .	69
5.3	Validación semántica del algoritmo . . . . .	120
5.3.1	Caso 1 . . . . .	121
5.3.2	Caso 2 . . . . .	122
5.3.3	Caso 3 . . . . .	124
5.4	Conclusiones de la validación . . . . .	126
<b>6</b>	<b>Evaluación</b>	<b>129</b>
6.1	Enfoques de evaluación . . . . .	129
6.1.1	Métodos cuantitativos de evaluación. . . . .	129
6.1.2	Los métodos cualitativos . . . . .	132
6.1.3	Discusión de los enfoques de evaluación . . . . .	132
6.2	Composición de la prueba . . . . .	133
6.3	Definición de datos reales . . . . .	134
6.4	Resultados de la evaluación . . . . .	140
6.5	Conclusiones de la evaluación . . . . .	146
<b>7</b>	<b>Conclusiones y trabajo Futuro</b>	<b>147</b>
7.1	Conclusiones . . . . .	147
7.2	Líneas futuras de investigación . . . . .	152
	<b>Bibliography</b>	<b>153</b>

# CAPÍTULO 1

---

## Introducción

---

La ciencia y tecnología informática se vincula con innumerables áreas del conocimiento, un ejemplo es la ingeniería inversa, la cual se beneficia de la relación del área de visión por computador y de las tecnologías semánticas, y es el punto de partida para la investigación con la que se da inicio a la presente tesis doctoral. La ingeniería inversa se define por Chikofsky and Cross (1990) como el proceso de análisis de un sistema con el objetivo de 1) identificar sus componentes e interrelaciones y 2) crear representaciones del mismo en otra forma o en un nivel superior de abstracción. Su área de aplicación es extensa representando diversos sistemas que van desde software (Eilam, 2011; Matzko et al., 2002), protocolos de comunicación (Cui et al., 2007; D. Lee and Sabnani, 1993; Shevertalov and Mancoridis, 2007), componentes electrónicos (Kumagai, 2000; Nohl et al., 2008), hasta sistemas industriales (Onuh et al., 2006; Otto and Wood, 1998; Yao, 2005). Dentro de este último campo, la ingeniería inversa juega un papel importante en el área de la creación de los inventarios; especialmente en la actualización de los planos existentes de las fábricas, que carecen de la información actualizada de los objetos reales. Estos planos pueden ser representados mediante técnicas de reconstrucción 3D.

La reconstrucción 3D es un problema importante en sí mismo y con frecuencia es usada como una fase intermedia en la meta global del reconocimiento de objetos en escenas 3D, tal es el enfoque principal de esta tesis. Cabe señalar que autores como Granholm (2015); Márton and Türker (2013); Prince (2012); Szeliski (2010) aseguran que el reconocimiento de objetos dentro de una escena 3D sigue siendo una de las tareas más complejas dentro del campo de visión por computador. En dicho campo se han desarrollado diferentes investigaciones para el estudio del reconocimiento de objetos, primero en imágenes 2D y recientemente en escenas 3D descritas en el estado del arte de esta tesis.

Las escenas 3D se obtienen mediante modernos dispositivos de rango, éstos realizan un detallado escaneo de objetos complejos, produciendo modelos de puntos 3D (a menudo denominado nubes de puntos). Una nube de puntos se define como un conjunto de puntos no organizados que representan en una forma discreta el objeto en cuestión y se caracterizan por sus coordenadas espaciales (Erdos et al., 2014). Se dice que son no organizados debido a que no poseen información adicional como topología, orientación o un punto de referencia, sin embargo, la mayor parte de la información se puede deducir mediante la aplicación de algoritmos adecuados (Sankaranarayanan et al., 2007).

La tecnología ha avanzado hasta ser capaz de reconstruir escenas 3D con precisión, no obstante todavía no es fácil de detectar y reconocer todos los objetos en ellas (Szeliski, 2010). La presente investigación se enfoca en el reconocimiento de primitivas geométricas que describan elementos comunes dentro de una escena industrial, específicamente cilindros, toros y esferas pertenecientes a una nube de puntos 3D, con lo que se pretende reducir las carencias e incrementar la efectividad de reconocimiento con respecto a los métodos actuales analizados en el estado del arte. El objetivo principal de esta investigación, consiste en proporcionar una maqueta digital enriquecida con alto grado de detalle, ofreciendo información no solo lógica, sino física y semántica.

## 1.1 Motivación de la investigación

Si bien, en la últimas décadas ha habido grandes avances al analizar una escena virtual reconociendo todos los objetos que aparecen en ella, esta tarea sigue siendo una de las más complejas de la visión por computador (Granholt, 2015; Márton and Türker, 2013; Prince, 2012; Szeliski, 2010). A pesar de la gran capacidad que se ha logrado al desarrollar escáneres para generar maquetas digitales precisas de instalaciones industriales (Lin et al., 2014; Marshall and Stutz, 2011), para el tratamiento y reconocimiento automatizado de elementos, dichas capacidades resultan irrelevantes ya que estas maquetas carecen de un significado semántico y de información relevante que las describa; es decir, las personas podemos percibir por medio de la vista los diferentes elementos que la componen una escena 3D, pero computacionalmente es imposible obtener información valiosa sin un algoritmo que reconozca los elementos y proporcione datos precisos y descriptivos. En este sentido, el reto es describir el entorno que percibimos en una escena virtual de modo que se detallen sus propiedades tales como tipo, forma y distribución. Esta investigación resulta relevante en el campo industrial en los casos donde los planos del interior de una fábrica son inexistentes, incompletos y/u obsoletos, por lo que surge la necesidad de contar con un mecanismo que genere una escena virtual enriquecida, además tiene la capacidad de extrapolarse a otros dominios.



## 1.2 Justificación

Describir detalladamente una escena digital, tal y como la percibe un humano no es una tarea fácil, mucho menos cuando los objetivos de esta descripción sean procesos pre-definidos como lo es la creación de inventarios. Un inventario es una relación detallada, ordenada y valioso de los elementos que componen los activos de una empresa o persona (Gaither and Frazier, 2000). Es *detallado* porque se especifican las características de cada uno de los elementos que integran un patrimonio. Por otra parte, se considera *ordenada* porque agrupa los elementos en sus respectivas cuentas. Y se *valora* porque el valor de cada activo se expresa en unidades. Esta tarea requiere de muchos factores como son conocimiento previo, capacidad de análisis y reconocimiento. Factores que no es posible encontrarlos en una sola ciencia, de esta manera, la hibridación del uso de maquetas digitales, la ingeniería inversa y tecnologías semánticas puede mejorar el proceso.

Como se detallará en el apartado de estado del arte, en la actualidad existen trabajos de investigación que se enfocan en este ámbito como los desarrollados por As Ari et al. (2014); Attene, Falcidieno, et al. (2006); Babinec et al. (2012); Biasotti et al. (2003a); Dornbusch et al. (2007); Gao et al. (2010); R. Rusu and Marton (2009); Tombari et al. (2011), por mencionar algunos. A pesar del esfuerzo desarrollado en estas investigaciones siguen existiendo algunas carencias como:

1. Manejo eficiente de nubes de puntos de grandes dimensiones.
2. Manejo eficiente de nubes de puntos con densidad variable.
3. Porcentajes de pérdida de información altos.
4. Porcentajes de clasificación errónea altos.
5. Tiempo de procesamiento bajo.

La presente investigación intenta solventarlas por medio de un algoritmo de reconocimiento de objetos que adopta los principios de análisis de descriptores de características geométricas y reglas semánticas para la representación de maquetas digitales enriquecidas con el objetivo de proporcionar información lógica, física y semántica de objetos representados en ella.

El uso de descriptores de características geométricas dentro de la nube de puntos ayudará a resolver la pregunta *¿Existe algún objeto conocido dentro de la escena escaneada?*. Mientras que el uso de tecnologías semánticas intentará resolver la pregunta *¿Qué tipo de elementos tiene la escena escaneada?*. De tal manera que se genere una maqueta digital enriquecida con **información lógica** calculando para cada objeto sus características geométricas tales como la altura, el ancho, el perímetro, la radio, por nombrar algunas; **información semántica** a partir del poblado de la ontología propuesta con la información de cada objeto; e **información física** mediante el cálculo de sus relaciones espaciales.

### 1.3 Objetivo general y objetivos específicos

En términos generales, el aporte principal de esta tesis es el reconocimiento de elementos a partir de la acotación semántica de primitivas geométricas básicas en una nube de puntos que describan elementos comunes dentro de una escena, específicamente cilindros, toros y esferas. Dicho reconocimiento considera como parámetros del modelo descriptores de características geométricas y reglas semánticas.

#### 1.3.1 Objetivo General

El objetivo general de la tesis es definir un algoritmo de ingeniería inversa, capaz de reconocer elementos a partir de primitivas geométricas, específicamente cilindros, toros y esferas en una nube de puntos captados vía un escáner de escenas de medio-gran tamaño y asociarlos lógicamente, semántica y físicamente con elementos que se pueden encontrar en una escena real. Esto permitirá la generación de una maqueta digital enriquecida en donde los objetos serán descritos con un alto grado de detalle, proporcionando no solo la forma, sino también características geométricas como alto, largo, diámetro; sus relaciones espaciales con otros objetos e información semántica de acuerdo al rol que representa cada elemento en la escena.

#### 1.3.2 Objetivos específicos

La definición y organización de los objetivos específicos es importante para el cumplimiento del objetivo general planteado. A continuación, se proporcionan los objetivos específicos a seguir durante esta investigación:

- El algoritmo debe estar conformado por módulos funcionales correctamente identificados y acotados.
- Identificar y analizar el formato de la fuente de datos proporcionada por el escáner, es decir, identificar que tipo de datos provee el escáner, si éste sigue algún formato o estándar a partir del cual se debe interpretar la información, de manera que permita seleccionar el mejor medio para gestionar la información.
- Incluir descriptores geométricos, los cuales representen y describan matemáticamente las primitivas a identificar.
- Diseñar y desarrollar el modelo semántico de objetos reales asociado a las primitivas encontradas en la nube de puntos, de manera que permitan asociar las entidades u objetos, con texto legible y entendido por humanos.
- Definir reglas semánticas que permitan representar fiel y semánticamente a los elementos pertenecientes a una escena real. Las reglas semánticas permitirán la correcta clasificación de los elementos considerando aspectos conocidos a priori como

son: el comportamiento de los algoritmos de segmentación de primitivas, así como la actuación de los objetos en el mundo real.

- Validar del algoritmo.

Cada uno de los objetivos específicos mencionados tiene la finalidad de servir de instrumento para finalizar de manera exitosa la investigación de esta tesis, requieren del uso de análisis matemático, tecnología semántica y de visión por computador.

## 1.4 Organización de la tesis doctoral

Esta tesis inicia proporcionando una descripción general del contexto en el que se identifica la problemática a resolver, seguido por la motivación de la investigación, así como los objetivos generales y específicos. Posteriormente en el capítulo 2 se proporciona el estado del arte relacionado con el ámbito del tema, el cual está dividido en dos secciones: la primera se enfoca en el reconocimiento de objetos 3D, mientras que la segunda hace referencia al enfoque semántico. En el tercer capítulo se plantean las hipótesis a resolver y su proceso de validación. Mientras que en el Capítulo 4, se describe el algoritmo con todas sus características y procesos para el reconocimiento y acotación semántica de objetos en una nube de puntos 3D. El Capítulo 5, incluye información detallada de los experimentos que permiten la validación de las hipótesis planteadas, y finalmente en el Capítulo 6, se proporcionan las conclusiones generales de la presente tesis, así como sus líneas futuras de investigación.



## CAPÍTULO 2

---

### Estado del Arte

---

En este capítulo se presenta el contexto que enmarca la presente investigación, para lo cual se describirán los conceptos fundamentales, tecnología relacionadas con la tesis y se comparan los principales trabajos relacionados seleccionados durante la revisión de la literatura. Este capítulo está dividido en dos secciones, la primera se enfoca en el reconocimiento de objetos 3D, para lo cual es necesario presentar la importancia del uso de descriptores visuales y la descripción del proceso general para el reconocimiento de objetos en nubes de puntos 3D, finalmente se presentan los enfoques del reconocimiento de objetos más relevantes seleccionados durante la revisión de la literatura. Una vez presentado las investigaciones actuales y las carencias de las mismas en relación a los objetivos de esta tesis, se presenta la segunda sección del estado del arte, es decir, el enfoque semántico que consiste en información teórica y técnica de tecnologías semánticas, la descripción de la web semántica, ontologías, metodologías ontológicas y lenguajes de implementación de ontologías, así como los trabajos relacionados en el contexto de tecnologías semánticas dentro del reconocimiento de objetos.

### 2.1 Reconocimiento de objetos 3D

La tarea de reconocer objetos en una escena ha sido estudiada con anterioridad en el campo de la Visión por Computador, la cual es una disciplina que estudia cómo procesar, analizar e interpretar imágenes de forma automática (Gonzalez et al., 2004). Dentro de este campo, el reconocimiento de objetos ha ido evolucionando, en primera instancia analizando objetos dentro de imágenes 2D y recientemente en escenas 3D (Szeliski, 2010). Esta tesis busca crear un algoritmo de reconocimiento de objetos 3D en una nube de puntos que bajo

el contexto del reconocimiento de objetos 3D resuelva la pregunta *¿Existe algún objeto conocido dentro de la escena escaneada?*.

En el contexto de visión por computador, un objeto es una entidad del mundo real con una etiqueta única. El significado de ese objeto depende del nivel de abstracción que se usa en contexto de la aplicación (Szeliski, 2010).

El reconocimiento de objetos es en la actualidad el mayor reto de la visión por computador, debido a esto, en las últimas décadas grandes esfuerzos se han concentrado en este tema, particularmente en conjuntos tridimensionales, los cuales suelen encontrarse en un entorno real.

Granholt (2015); Márton and Türker (2013); Prince (2012); Szeliski (2010) reportan que en la tarea del reconocimiento de objetos no se ha logrado alcanzar las capacidades similares a las de un niño de 2 años, entre las que podemos mencionar: contar animales diferentes o reconocer un adulto familiar existente en una fotografía, reconocer objetos parcialmente ocultos, reconocer objetos con cambios estructurales, hacer coincidir objetos grandes y pequeños, ensamblar 3 o 4 piezas de un puzzle, entre otras (Lam, 2005). A su vez, algunos autores coinciden que la problemática se encuentra en delimitar el concepto de “objeto” dentro del proceso, lo que da origen a los enfoques generales en los que se afrontan los problemas de reconocimiento. En Szeliski (2010) se mencionan 4 tipos de reconocimiento:

- **Detección de objetos.** Es utilizada en casos donde el objeto a detectar es conocido y no se mueve. Entonces, un proceso de barrido a la imagen o escena es utilizado. Este proceso tiene el objetivo de determinar en qué parte de la fuente se produce una correspondencia con el objeto modelo.
- **Reconocimiento de instancias.** Consiste en el reconocimiento de un objeto rígido (en 2D o 3D) conocido, ubicado en un fondo desordenado, y con oclusiones parciales. El objetivo del reconocimiento de instancias es buscar la correspondencia (reconocer) un objeto o escena específico. Los objetos deben ser reconocidos sin importar cambios de escala el punto de vista de la cámara, condiciones de iluminación incluso oclusiones parciales.
- **Reconocimiento de categorías.** Este tipo de reconocimiento afronta problemas donde es necesario reconocer cualquier instancia de una clase general, por ejemplo un objeto de la clase “casa”, “avión” o “animal” (Ponce et al., 2007). En algunos casos, el proceso de reconocimiento de categorías se apoya en el contexto de los objetos circundantes y elementos de la escena.
- **Comprensión de contexto y de escena.** Es de más alto nivel. Este tipo de reconocimiento considera la comprensión de la escena (contexto) en el que el objeto está teniendo lugar. Tomar en cuenta el contexto puede mejorar considerablemente

el rendimiento de algoritmos (Divvala et al., 2009), así como proporcionar pistas semánticas útiles para comprender la escena general.

Considerando todo lo anterior, el tipo de abstracción de objetos que se desea utilizar en la presente tesis es a nivel de comprensión de contexto y de escena. Para abordar por completo el tema del reconocimiento de objetos en una escena, a continuación, se detallan diferentes aspectos generales tales como los descriptores visuales, el proceso general para el reconocimiento de objetos en nubes de puntos y los diferentes enfoques existentes seleccionados a partir de la revisión de la literatura.

### 2.1.1 Proceso general de reconocimiento de objetos en nube de puntos

#### 2.1.1.1 Descripción de subprocesos

Reconocer un objeto conlleva el desarrollo de diferentes subprocesos en un estricto orden, el proceso general de reconocimiento de objetos en una nube de puntos 3D descrito en la Figura 2.1.1 refleja los diversos subprocesos que se emplean comúnmente. Para el óptimo desarrollo de un proceso de reconocimiento de objetos, es necesario considerar los subprocesos de captura, pre-proceso, definición de descriptores, clasificación, verificación e interface.

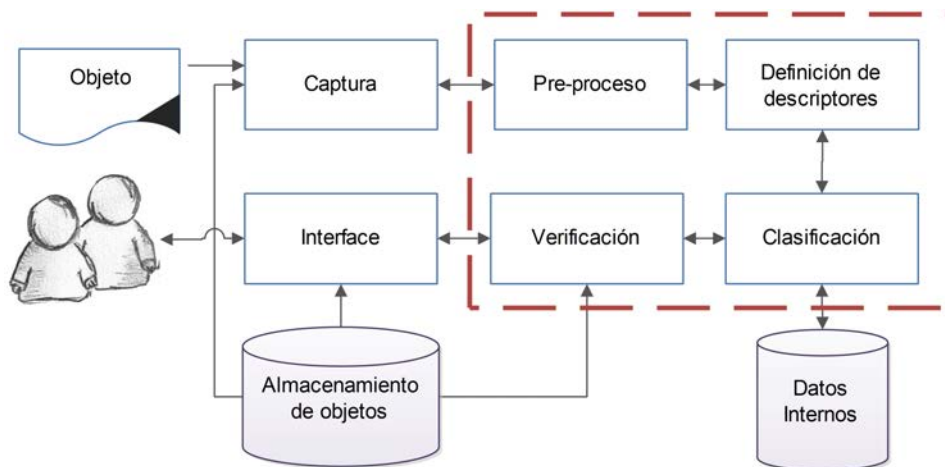


Fig. 2.1.1: Proceso general para el reconocimiento de objetos

Dichos subprocesos para el reconocimiento de objetos son detallados a continuación:

- **Captura.** La captura 3D es un método mediante el cual una superficie se muestrea o escanea usando tecnología láser. El objeto o escena a capturar es analizado para obtener información sobre su forma y su apariencia. La ventaja de la captura 3D por medio del escaneado láser es que es posible obtener una gran cantidad de puntos

con alta precisión en un periodo de tiempo relativamente corto. Debido a que los escáneres láser son instrumentos de línea de vista, es necesario realizar múltiples capturas desde diferentes estaciones para garantizar una cobertura completa de un objeto o escena (Marshall and Stutz, 2011).

- **Pre-proceso.** El objetivo del pre-proceso es reducir toda la información mejorando la calidad y/o la apariencia de la imagen o escena original (Gonzalez, 2009). Se resaltan ciertas características de una imagen como bordes y contraste, por nombrar algunos; y se ocultan o elimina la información irrelevante, por ejemplo el ruido. El pre-procesamiento es la etapa previa a la fase del análisis en la cual se abordan aspectos como segmentación, extracción de características, reconocimiento e interpretación (Bieniecki, 2007). Los principales tipos de pre-proceso son realizado y filtrado.
- **Definición de descriptores.** El objetivo de la extracción de descriptores es reducir al mínimo la cantidad de información esencial para detectar objetos. Tal y como se ha mencionado, para reconocer un objeto es necesario extraer características que permitan representarlo y describirlo matemáticamente (Cabral et al., 2005). Un conjunto de descriptores de mala calidad produce un solapamiento de clases y por lo tanto una alta probabilidad de error en la clasificación.
- **Clasificación.** La clasificación es el proceso de convertir los datos brutos en información significativa, útil y comprensible (Mountrakis et al., 2011). En esta etapa se decide la categoría más probable a la que pertenece cada entidad considerando su vector de propiedades (Diaz et al., 2004). La clasificación tiene un planteamiento matemático bien definido y puede considerarse resuelta desde el punto de vista de la Teoría de la Decisión. Cuando el conjunto de propiedades obtenidas en la etapa anterior (extracción de características) es suficientemente discriminante, la complejidad de esta etapa se reduce (Jia and Richards, 1999). El diseño correcto del clasificador contribuirá a disminuir la proporción de errores.
- **Verificación.** El objetivo es la evaluación de los resultados del clasificador. La verificación puede incluir alguna información extra de fuentes adicionales al sistema, incluso información proporcionada por el usuario.
- **Interfaz.** Comunicación entre el proceso y el usuario. Tiene el objetivo de desplegar los resultados obtenidos en los procesos anteriores, a través de la pantalla.

### 2.1.2 Descriptores Visuales

Un descriptor visual se define como la descripción detallada de un objeto, el cual es considerado para su identificación o análisis en un proceso de reconocimiento de objetos (J. Wu and Rehg, 2011). Un descriptor visual está formado por un conjunto de datos



ordenados, que contiene información acerca del objeto que describe y suelen presentarse en forma de vector o matriz. La imagen o escena digital, en sí misma, puede considerarse un descriptor visual. Un descriptor visual debe poseer las siguientes propiedades:

- **Simplicidad:** El descriptor debería representar las características extraídas de la escena de manera clara y sencilla para permitir una fácil interpretación de su contenido.
- **Repetibilidad:** El descriptor generado a partir de una escena debe ser independiente del momento en el que se genere.
- **Diferenciabilidad:** Dada una escena, el descriptor generado debe poseer alto grado de discriminación respecto de otras escenas y al mismo tiempo contener información que permita establecer una relación entre escenas similares.
- **Invarianza:** Cuando existen deformaciones en la representación de dos escenas, es deseable que los descriptores que las representan aporten la robustez necesaria para poder relacionarlas aún bajo diferentes transformaciones.
- **Eficiencia:** Es deseable que los recursos consumidos para generar el descriptor sean aceptables para poder ser utilizados en aplicaciones con restricciones críticas de espacio y/o tiempo.

De acuerdo a la diversidad de los descriptores visuales, es difícil clasificarlos según su naturaleza, a pesar de que casi todos se fundamentan en la evaluación de alguna característica diferenciadora del objeto como intensidad, color, forma etc. Sin embargo, de acuerdo a su nivel de abstracción los descriptores visuales se pueden catalogar en dos grandes grupos:

- Descriptores visuales globales
- Descriptores visuales locales

A continuación se describe de forma más detallada cada uno de ellos.

### 2.1.2.1 Descriptores Visuales Globales

Los descriptores globales resumen el contenido de la imagen o escena en un único vector o matriz de características, tienen la ventaja de agrupar una gran cantidad de información de la muestra en tan solo una pequeña cantidad de datos para describirla. A pesar de su simplicidad, este tipo de descriptores han resultado ser ampliamente utilizados para diferentes tareas, debido entre otras cosas a su bajo coste computacional. Los descriptores globales más relevantes son detallados a continuación.

**Descriptores visuales basados en vectores normales.** Para la comprensión de este tipo de descriptores, es necesario definir el concepto *normal*. La normal a una superficie en un punto  $P$  es un vector perpendicular al plano tangente a la superficie en  $P$ . El concepto de normalidad se generaliza a la ortogonalidad, esto es, el espacio de vector normal es el conjunto de los vectores que son ortogonales al espacio tangente en  $P$ . La normal se utiliza para determinar la orientación de una superficie hacia una fuente de luz para el sombreado plano, o la orientación del punto dado para imitar una superficie curva (Blinn, 1978). Debido a la información que proporcionan los vectores normales acerca de las superficie de los objetos, se han desarrollado descriptores basados en ellos, entre los más destacados se encuentra VFH (*Viewpoint Feature Histogram*), el cual describe un histograma calculado para cada imagen o escena, en donde se captura tanto la forma del objeto como el punto de vista desde donde se captura la imagen. Para almacenar el modelo completo de un objeto, es decir, la forma del objeto y sus diferentes puntos de vista, es necesario almacenar un descriptor correspondiente a cada punto de vista. El descriptor VFH ha sido ampliamente utilizado, por ejemplo Babinec et al. (2012) desarrollaron una mejora al mismo con el objetivo de guiar un robot móvil para evitar obstáculos dinámicos y seleccionar el camino más adecuado. El algoritmo emplea la reducción de datos múltiples que resulta en un histograma polar, el cual es dividido en sectores angulares alrededor de donde se encuentra el robot, por medio del histograma se analiza si se trata de una apertura o un espacio bloqueado. Otro descriptor basado en normales es CVFH (*Clustered Viewpoint Feature Histogram*), el cual fue presentado por Aldoma et al. (2011). CVFH es una extensión de VFH, basada en la idea de que los objetos tienen una cierta estructura que permite dividirlo en un número determinado de regiones finas, cada una de estas regiones finas se utiliza para calcular de forma independiente un conjunto de histogramas VFH. CVFH emite una representación multivariante del objeto de interés, debido a esto, este descriptor es más robusto para los artefactos de segmentación. Por otra parte, CSHOT (*Color- Signature of Histograms of Orientations*) es un enfoque propuesto por Tombari et al. (2011), el cual se deriva de la teoría de un descriptor SHOT para los datos 3D que se basa solamente en forma de los objetos. El descriptor CSHOT mejora la precisión de la función de adaptación en escenarios de reconocimiento de objetos caracterizados por la presencia de desorden y oclusiones.

**Descriptores visuales basados en vóxel.** Los datos gráficos se representan comúnmente en una de tres formas principales: 1) como un modelo de objeto continuo (geométrica), como una imagen en mapa de bits discreta pixel-imagen, y como un espacio discreto voxel-imagen (Kaufman and Shimony, 1987). Entonces, un voxel es la representación volumétrica de los objetos (Steinbach and Girod, 2000). La representación de objetos 3D que permite el uso de mapas de voxels ha inspirado el desarrollo de algunos descriptores como SDVS (*Shape Distributions on Voxel Surfaces*) es un descriptor presentado por Wohlking and Vincze (2011), el cual es capaz de clasificar un “objeto nunca antes visto” en su primera aparición. Este descriptor de forma utiliza las distribuciones

de superficies en voxels y se puede utilizar en diversas aplicaciones: como un descriptor 3D puro para la recuperación de modelo 3D o como un descriptor 2.5D para encontrar modelos 3D a vistas. A su vez, ESF (*Ensemble of Shape Functions*), fue desarrollado por Wohlkinger and Vincze (2011). ESF es un descriptor de forma de nubes de puntos parciales a base de funciones de forma. La clasificación se presenta por medio de la recuperación 3D al encontrar los vecinos más cercanos de visitas generadas sintéticamente de modelos CAD a la nube de puntos detectada. El descriptor de forma muestra que la combinación de las funciones de ángulos, distancias de puntos y área de las formas da un impulso significativo en la tasa de reconocimiento. Por último, GRSD (*Global Radius-Based Surface Descriptor*) es una versión global del descriptor RSD, el cual es presentado por Marton and Pangercic (2010), GRSD trabaja en dos etapas: 1) la anotación de grupos de puntos con categorías basadas en su geometría, tales como: cajas, bandejas, platos, etc, donde para cada célula de voxel es utilizado el descriptor de superficie basada en el radio (RSD) y luego 2) usando apariencia basada en características visuales (descriptor SURF, el cual se describirá más adelante) para obtener el objeto exacto y su orientación con respecto al eje vertical del objeto.

**Descriptores visuales con un dominio específico.** Descriptores geométricos, como los presentados por Lin et al. (2014) en donde se clasifican nubes de puntos de zonas urbanas obtenidas con un escáner LIDAR aerotransportado. El objetivo principal es generar eigen-características fiables de una nube de puntos para mejorar la precisión de la clasificación. El método se basa en la matriz de covarianza ponderada con la mediana geométrica, que es insensible a la naturaleza no uniforme de las muestras propias de una exploración LIDAR. En el análisis de cada eigen-característica se le asigna un peso a cada punto para representar la contribución espacial del punto en el análisis de componentes principales ponderados y para estimar la mediana geométrica que puede ser considerada como el centro de la forma. Por otro lado se encuentran los descriptores de escena entre los cuales se puede mencionar J. Wu and Rehg, 2011, el cual es un descriptor visual para reconocer lugares topológicos o categorías de escena llamado CENTRIST (CENsus TRansform hISTogram), el cual analiza principalmente las propiedades estructurales dentro de una imagen y suprime la información textural detallada. Por último se presentan los descriptores de forma, los cuales son un conjunto de herramientas matemáticas y procedimentales que permiten obtener diversas medidas sobre los contornos y estructuras morfológicas de modelos o imágenes 3D. Los descriptores de forma existen en diferentes tipos: los que están basados en modelos de superficies cuadráticas; los que están basados en curvaturas principales, los que están basados en transformaciones de información 3D a 2D como es el caso de las imágenes spin que pueden ser calculadas a diferentes resoluciones, los multi-escala, entre otros (Lowe, 1999; Lowe, 2004; McAdams et al., 2011).

### 2.1.2.2 Descriptores Visuales Locales

Actúan sobre regiones de interés, previamente calculadas o identificadas, construyendo un vector de características de esa región que tiene en cuenta la información contenida tanto en el punto de interés como en la región adyacente al mismo o vecindario. Normalmente las regiones descritas se conocen como puntos de interés (en reconocimiento basado en 2D), también llamados puntos destacados o keypoints (en reconocimiento basado en 2D y 3D). El descriptor entonces, está constituido por la totalidad de los vectores de características calculados. Los descriptores locales más relevantes se describen a continuación.

**SIFT** (*Scale-Invariant Feature Transform*) es un descriptor local que tiene la capacidad de identificar con firmeza objetos incluso entre el desorden y bajo oclusión parcial, porque la descripción de la característica SIFT es invariante a la escala, orientación e iluminación. Primero los puntos significativos SIFT se extraen de un conjunto de imágenes de referencia y se almacenan en una base de datos. Un objeto se reconoce en una nueva imagen mediante la comparación individual de cada característica de la nueva imagen con las que están en la base de datos; el porcentaje de similitud se mide por medio de la distancia euclídea de sus vectores de características. Este descriptor es ampliamente utilizado en investigaciones actuales como Y. Li, W. Liu, et al. (2014); Liao et al. (2013); Sykora et al. (2014); X. Wu et al. (2014) Así mismo se puede mencionar a SURF (*Speeded Up Robust Features*), el cual fue presentado por primera vez por Bay et al. (2006). El descriptor **SURF** se puede utilizar en tareas de reconocimiento de objetos o reconstrucción 3D. SURF calcula características locales, distintivas e invariantes rápidamente. Está inspirado en el descriptor SIFT y es más rápido debido a variaciones en deformaciones tales como cambios de escala, rotación de imagen, los cambios de punto de vista, las imágenes borrosas, cambio compresión JPEG y cambios de luz. SURF es computacionalmente eficiente y resistente a la oclusión parcial. Desde su creación ha sido ampliamente utilizado en diferentes investigaciones tales como Kang et al. (2015). Por último se presenta **FAST** (*Features from Accelerated Segment Test*). FAST es un algoritmo propuesto originalmente por Rosten and Drummond (2006). El algoritmo llamado Características de la prueba de segmento acelerado (FAST) es un método de detección de esquina, es utilizado para extraer características y además para rastrear objetos. La mayor ventaja de este descriptor es su eficiencia computacional. FAST es adecuado para aplicaciones que requieran un rendimiento de alta velocidad (Mair et al., 2010).

### 2.1.2.3 Discusión del proceso de reconocimiento

Una vez presentado el concepto de descriptores visuales, sus características y tipos, es necesario describir el proceso general de reconocimiento de objetos en una nube de puntos en donde los descriptores visuales tienen un papel importante.

El reconocimiento de objetos se basa comúnmente en el diseño y uso de modelos de los objetos, los cuales son previamente analizados y almacenados, dicha información es,

de acuerdo al contexto, particularmente seleccionada y tiene el objetivo de optimizar el proceso de reconocimiento. La selección de los descriptores visuales que definen un modelo digital, es uno de los pasos más críticos dentro del proceso de reconocimiento y depende de las características de los objetos que se pretende reconocer. Los descriptores visuales de los modelos pueden ser diseñados de forma automática o supervisada. El algoritmo de reconocimiento general es definido en sí por el proceso de comparación entre los modelos de objetos y los objetos de la fuente de datos a reconocer. En los últimos años ha habido muchos esfuerzos para optimizar el proceso de reconocimiento de objetos en nubes de puntos, sin embargo, existen investigaciones como As Ari et al. (2014); Feng et al. (2015); Ivorra et al. (2015); Jensfelt and Kragic (2006) que han enfocado sus esfuerzos en los subprocesos extracción de descriptores y clasificación aplicada a diferentes ámbitos. En este sentido y de acuerdo con los objetivos de la tesis, esta investigación está enfocada en los dos subprocesos más relevantes del reconocimiento de objetos: definición de descriptores y clasificación, pero adicionalmente abarcamos un subproceso antes (pre-proceso) y otro después (verificación) para asegurar la calidad de la clasificación de los objetos. Estos cuatro procesos serán descritos detalladamente más adelante en el capítulo 4.

### 2.1.3 Enfoques del reconocimiento de objetos

Como ya se ha mencionado, el problema de reconocimiento de objetos ha sido abordado por diferentes investigaciones. La literatura presenta algoritmos de reconocimiento de objetos 3D clasificados de acuerdo a su enfoque (Tangelder and Velkamp, 2008). A continuación se describen los más relevantes para esta investigación.

#### 2.1.3.1 Enfoque basado en grafos topológicos

Estos sistemas evalúan la estructura, forma y geometría de los objetos, prestando especial atención a cómo se encuentran conectadas las distintas partes de la geometría del objeto. El enfoque basado en grafos topológicos distingue y reconoce tanto formas y volúmenes como deformaciones flexibles (torcimientos y estiramientos). Los sistemas basados en grafos se dividen, a su vez, en dos categorías: grafos de modelo y esqueletos.

- **Basados en grafos de modelo.** Los sistemas basados en grafos de modelo se utilizan sobre todo para la recuperación de modelos generados con herramientas CAD. Entre estos sistemas destacan las representaciones de contorno, que describe un modelo en términos de vértices, lados y caras, y la geometría sólida constructiva (CSG), que describe un modelo en términos de un conjunto de operaciones booleanas aplicadas a entidades geométricas básicas tales como cubos o cilindros. Dentro de la literatura que abarca este enfoque se encuentra el trabajo descrito en Zuckerberger et al., 2002.

- **Basado en esqueleto.** Los cuales utilizan como descriptor de forma un grafo basado en esqueleto que codifica información geométrica y topológica. Tras someter al objeto a una transformación para obtener una estructura de vóxeles, los nodos del esqueleto se obtienen al aplicar un algoritmo de adelgazamiento basado en transformación de distancias. Posteriormente se conectan los nodos en un grafo de forma acíclico tras aplicar el algoritmo de árbol de mínima expansión, ejemplos de este enfoque son descritos en Biasotti et al. (2003a); Biasotti et al. (2003b); Hilaga and Shinagawa (2001); Mortara and Patané (2004); Steiner and Fischer (2002).

Este enfoque clasifica objetos compuestos completos, más no reconoce características detalladas de las superficies; tiene un bajo porcentaje de éxito en la clasificación de objetos cuando existen oclusiones parciales ya que deforma la topología; además que es computacionalmente costosa.

### 2.1.3.2 Enfoque basado en geometría

Los métodos basados en geometría localizan casos de interés en imágenes tridimensionales. Se dividen en modelos basados en vistas, en error volumétrico y los estructurales.

- **Modelos basados en vistas.** La idea principal subyacente en estos métodos es que dos modelos tridimensionales son similares si todas las vistas desde distintos ángulos de visión también son similares. Estos modelos facilitan el aprendizaje automático de los objetos de modo estable. Sin embargo en la práctica una comparación directa de las imágenes como mapas de bits no es óptima, ya que implica una carga computacional excesiva, y además este tipo de modelos son muy sensibles a ruidos, cambios de iluminación y otras alteraciones, además presentan una información no estructurada. Trabajos como Gao et al. (2010); Sung and Poggio (1998) se encuentran dentro de este enfoque.
- **Basados en error volumétrico.** Los cuales describen la similitud geométrica entre dos modelos tridimensionales definiendo el error volumétrico existente entre ambos mediante una función que cuantifica el número de vóxeles que han de desplazarse y la distancia que estos se desplazan para lograr dos formas prácticamente iguales. Dentro de los trabajos más relevantes en este enfoque se encuentran Cajal et al. (2013); Masood and Rattanawong (2002); Masood, Rattanawong, and Iovenitti (2000).
- **Modelos estructurales.** Dichos modelos descomponen cada objeto en un conjunto de figuras geométricas. El mayor problema de estas técnicas es que los modelos presentan una inestabilidad inherente, debido a que un mismo objeto se puede descomponer en distintas combinaciones primitivas. Ejemplos relevantes de este tipo de enfoques, es posible mencionar Attene, Falcidieno, et al. (2006); Attene and Patane (2010); Dey et al. (2003).

Los enfoques geométricos para el reconocimiento de objetos 3D se limitan al reconocimiento de formas simples y son sensibles a ruidos y a deformaciones flexibles, aunque soportan escenas desordenadas y a diferencia del enfoque basado en grafos no utilizan descriptores basados en la apariencia. Debido a las deficiencias de los modelos mencionados anteriormente, la presente investigación propone un algoritmo de reconocimiento de objetos 3D basado en el desarrollo de descriptores visuales, pero agrega tecnología semántica con el fin de solventar dichas deficiencias. En consecuencia, esta investigación hace uso de descriptores de características geométricas y reglas semánticas para obtener como resultado nubes de puntos 3d enriquecidas con información lógica, física y semántica. Por lo que, es evidente la necesidad de describir que tecnología semántica se utilizará en esta tesis.

## 2.2 Tecnologías semánticas

El principal aporte de esta tesis es el desarrollo del algoritmo de reconocimiento de objetos 3D, en el cual se propone el uso de descriptores visuales y de tecnologías semánticas. Por lo que en esta sección se incluye información teórica y técnica de tecnologías semánticas, a partir de las cuales permiten sustentar y completar el proceso de clasificación de objetos en una nube de puntos 3D que se describe en el Capítulo 3. El uso de tecnologías semánticas en la presente tesis intentará resolver la pregunta ¿Qué tipo de elementos tiene la escena escaneada?. Esta sección inicia con la descripción de la propia web semántica, ontologías, así como metodologías para el desarrollo de ontologías, lenguajes ontológicos y herramientas para su desarrollo.

### 2.2.1 La web semántica

La evolución que llevó a la web de disponer de documentos para ser leídos por humanos a incluir datos e información para que los ordenadores pudieran manipularlos, es decir la web semántica, fue descrita por primera vez por T. Berners-Lee (2001). La web semántica es una web de información procesable, en donde la teoría semántica proporciona un “*significado*” a dicha información permitiendo una interoperabilidad entre sistemas a través de la conexión lógica de términos (Shadbolt et al., 2006). Para lo cual la web semántica plantea la reestructuración y el enriquecimiento de documentos y componentes web con información semántica explícita, independiente de la presentación al usuario final.

Entonces, la web semántica es la extensión de la web tradicional (World Wide Web), la cual que permite compartir contenido más allá de los límites de las aplicaciones y sitios web. Una visión utópica, una red de datos, o simplemente un cambio de paradigma natural de la web tradicional, han sido algunas de las definiciones de la web semántica.

La Web tradicional está basada en el lenguaje HTML, el cual por medio de su más relevante característica de marcado permite a los buscadores clasificar los documentos HTML, crear un índice o catálogo y mostrarlo al usuario por medio del navegador. Son grandes las ventajas y beneficios que el HTML trajo a la web tradicional, sin embargo

carece de una completa descripción de datos y no es extensible. A partir de esas carencias, la web evolucionó con la introducción de otros elementos como DHTML, Javascript, e incluso lenguajes que permitían información más estructurada como lenguaje XML. No obstante este tipo de elementos y lenguajes no subsanaban la necesidad de una descripción más detallada del documento y de su contenido, además de no propiciar la interconexión entre sistemas. En consecuencia, surgieron lenguajes de marcado capaces de representar el conocimiento basándose en el uso de metadatos y ontologías. De acuerdo con Berners-Lee, la arquitectura de la web semántica se representa tal y como muestra la Figura 2.2.1:

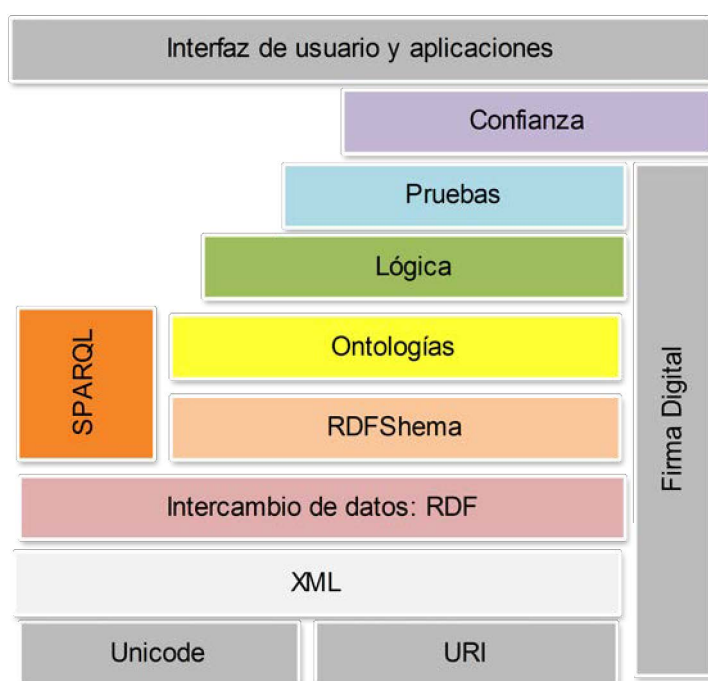


Fig. 2.2.1: Arquitectura de la web semántica. Adaptado de Berners-Lee (2006)

### 2.2.1.1 Ontologías

Una de las primeras definiciones de ontología es la descrita por Gruber (1993) en donde menciona que una ontología es una especificación explícita de una conceptualización. Para los sistemas basados en el conocimiento, lo que “existe” es exactamente lo que se puede representar. Dentro de un dominio específico, se le llama universo del discurso al conjunto de objetos que se puede representar. Mientras que un vocabulario es la descripción del conjunto de objetos y sus relaciones entre ellos. Por medio de este vocabulario de representación un programa basado en el conocimiento puede representar el conocimiento (N. Noy and McGuinness, 2001), por lo que, una base de conocimiento se constituye a partir



de una ontología y un conjunto de instancias individuales.

En una ontología las *definiciones* asocian: 1) **nombres** de entidades en el universo de discurso, es decir, clases, relaciones, funciones, u otros objetos, con texto legible y entendido por humanos; 2) **propiedades** de cada concepto que describen diversas características y atributos del mismo, llamados slots, roles o propiedades; y 3) **axiomas** formales que limitan la interpretación y el uso de estos términos bien formados (Gruber, 1993; N. Noy and McGuinness, 2001).

Debido a la importancia de esta tecnología semántica, en la literatura se han encontrado trabajos que utilizaron ontologías para representar conocimiento en distintos tipos de dominios, como el diseño estructural (Hou et al., 2015), biomedicina (Levine et al., 2015; Mugzach et al., 2015), sector energético (T. Hong et al., 2015; López et al., 2015), tendencias de los mercados (K. C. Lee et al., 2015), e incluso en el campo de la robótica y automatización (Dogmus et al., 2015; Fiorini et al., 2015).

A medida que la cantidad de información en formatos digitales aumenta, las ontologías se han vuelto más populares para representar y generar conocimiento (Dogmus et al., 2015). Una característica notable de las ontologías es que permiten la aplicación del conocimiento por nueva información, es decir, ontologías desarrolladas en diferentes investigaciones e instantes tienen la capacidad de ser integradas y el razonamiento puede ser automatizado sobre conceptos y sus relaciones representadas en todas las ontologías. Por lo que, no es de extrañar que existan cada vez mas sistemas basados en conocimiento que se basan en ontologías.

Dependiendo de su nivel de generalidad Fensel (2001); Guarino (1998); Heijst et al. (1997) distinguen cuatro tipos de ontologías:

- **Ontologías de dominio.** Aquellas ontologías que representan el conocimiento válido para un tipo particular de dominio, por ejemplo, el dominio de la medicina, robótica, biología, industrial, por mencionar algunas.
- **Ontologías metadatos.** Son aquellas que proporcionan un vocabulario para representar el contenido de las fuentes de información en línea.
- **Ontologías genéricas o de sentido común.** Se refiere a las ontologías que tienen como objetivo capturar el conocimiento general sobre el mundo, representando conceptos generales y fundacionales del conocimiento, proporcionando nociones y conceptos básicos. Por lo que, estos conceptos son válidos en varios dominios, es decir, una ontología industrial es aplicable en ámbitos técnicos o comerciales.
- **Ontologías de representación** Son aquellas que representan conceptos aplicables en cualquier dominio. Este tipo de ontologías recogen los primitivos semánticos que se utilizan para modelar el conocimiento en un paradigma de representación del conocimiento. Como primitivos semánticos se puede mencionar a las clases, las relaciones y los atributos. Una ontología representacional relevante es la *Frame*

*Ontology* (Gruber, 1993), la cual define conceptos como frames, slots, y restricciones de slots, mismos que permiten la representación del conocimiento de una manera orientada a objetos o basado en marcos.

- **Ontologías de tareas y ontologías de métodos.** Las ontologías de tareas son aquellas que proporcionan términos específicos para tareas particulares. Mientras que las ontologías de métodos proveen términos específicos para métodos particulares de resolución de problemas. Ambas proveen un punto de vista de razonamiento sobre conocimiento del dominio.

### 2.2.1.2 Metodologías para el desarrollo de ontologías

Actualmente existe una gran cantidad de metodologías para la construcción de ontologías, todas ellas dependen de la aplicación final que tendrá la ontología. Una breve descripción de las metodologías más relevantes se presenta a continuación.

#### **Metodología Cyc.**

La metodología básica de mapeo Cyc fué presentada por Lenat and Guha (1993). Cyc inicia como un proyecto de inteligencia artificial que tiene como objetivo la construcción de una ontología comprensible para el razonamiento humano (Reed and Lenat, 2002).

Los pasos generales para la construcción de ontologías de acuerdo a Cyc son:

1. Extraer manualmente el conocimiento común que está implícito en diferentes fuentes
2. Utilizar herramientas de procesamiento de lenguaje natural o aprendizaje natural para la adquisición de nuevo conocimiento en la ontología.

El proyecto Cyc surgió por parte de la Corporación de Tecnología en Computación y Microelectrónica. La base de conocimiento de Cyc es propietaria, aunque una pequeña versión fue liberada y está disponible como OpenCyc, la misma que busca definir un vocabulario común para el conocimiento automatizado (Corcho et al., 2003).

#### **Metodología de Uschold y King.**

Metodología presentada por Uschold and King (1995) y basada en la experiencia de desarrollo de *Enterprise Ontology*. Dicha ontología fué diseñada para los procesos de modelado empresarial. Debido a la experiencia adquirida, la metodología de Uschold y King proporciona directrices para ontologías en desarrollo y permite la creación de ontologías en base a otras ya existentes (Corcho et al., 2003).

Los autores recomiendan los siguientes pasos:

1. Identificación del propósito de la ontología. Es esta primera etapa se debe tener claro el objetivo y usos al que se destinará la ontología.

2. Capturar los conceptos y las relaciones en el dominio de interés. Tiene el objetivo de definir el alcance de la ontología, enfocándose en los conceptos útiles para el dominio y sin ambigüedades.
3. Codificación de la ontología. Implica la representación explícita y correcta de los conceptos elegidos en el paso anterior en un lenguaje formal. Es posible la integración de ontologías existentes durante los procesos de captura y codificación.
4. Evaluación de la ontología.
5. Documentación de la ontología.

### **Metodología de Grüninger y Fox.**

Metodología basada en la experiencia obtenida a partir del desarrollo de la ontología del proyecto TOVE, la cual fue diseñada por Grüninger and Fox (1995) dentro del dominio de los procesos de negocio y actividades de modelado (Corcho et al., 2003). Esta ontología se enfoca en la construcción de un modelo lógico del conocimiento, el cual no se construye directamente sino en pasos: se inicia con la definición de una descripción informal que cubra las especificaciones de la ontología y después ésta se formaliza.

Los pasos para la construcción de ontologías de acuerdo con Grüninger y Fox son:

1. Definición de escenarios motivadores. Tiene como objetivo definir los escenarios que se plantean en la aplicación de la ontología. Cualquier propuesta de una nueva ontología o la extensión a una existente debe definirse en escenarios motivadores. Cada escenario motivador proporciona también un conjunto de posibles soluciones a los problemas de planteados en ellos.
2. Formulación de preguntas en lenguaje natural. Se les denomina preguntas de competencia, el objetivo del desarrollo de estas preguntas es determinar el ámbito de la ontología. Son basados en los escenarios motivadores ya definidos y son considerados como requisitos expresividad. La ontología a desarrollar debe ser capaz de responder a estas preguntas utilizando su terminología, y tener la capacidad de caracterizar las respuestas usando sus axiomas y definiciones.
3. Especificación de la terminología. En base a las preguntas realizadas en el paso anterior, se especifica la terminología de la ontología utilizando un formalismo. Estos términos darán pie a las definiciones, relaciones, propiedades y limitaciones que serán expresados por medio de axiomas.
4. Formalización de las interrogantes. Se definen formalmente las preguntas utilizando la terminología de la ontología.

5. Especificación de axiomas formales. Los axiomas son detallados como frases de primer orden para definir los términos y las limitaciones de los objetos en la ontología. Una ontología válida no es un conjunto de objetos, términos o conceptos, éstos deben estar acompañados de axiomas que definen la semántica o significado, de estos términos.
6. Verificación de la ontología. Se determinan las condiciones para evaluar la integridad de la ontología, es decir, se definen las condiciones en las cuales las soluciones a las preguntas estén completas.

Esta metodología basa su desarrollo en la correcta formulación de las preguntas de competencia ya que la ontología estará dirigida a responderlas.

#### **Metodología Methontology.**

Es una metodología creada por Gómez-Pérez (1998) con el objetivo de construir ontologías nuevas, reutilizando otras o a través de un proceso de reingeniería. Methontology presenta un ciclo de vida basado en la evolución de prototipos, siguiendo el estándar IEEE 1074 de desarrollo de software (Corcho et al., 2003; Gómez-Pérez et al., 2006).

Los autores sugieren los siguientes pasos para el desarrollo de la metodología:

1. Especificación. Definir el alcance y finalidad de la ontología expresada en lenguaje natural.
2. Conceptualización. Definición de los conceptos, casos, relaciones y propiedades mediante una representación informal.
3. Implementación. Representa la formalización de la ontología, en un lenguaje formal como RDF, OWL, por mencionar algunos.
4. Evaluación. Tiene el objetivo de comprobar el correcto funcionamiento de la ontología mediante pruebas verificando posibles inconsistencias y redundancias.

#### **2.2.1.3 Lenguajes de implementación de ontologías**

A continuación se presentan los lenguajes ontológicos más relevantes utilizados en la web. Algunos de estos lenguajes existían antes y fueron adaptados para su uso en la web, ahora son utilizados como lenguajes intermedio entre sistemas, mientras que otros lenguajes han sido diseñados con el propósito específico de la comunicación via web.

Un lenguaje ontológico debe cumplir con las siguientes características:

1. Debe permitir representar el conocimiento ontológico de manera sencilla
2. Debe tener una semántica axiomatizable

3. Debe permitir el razonamiento que necesitarán los agentes
4. El coste computacional del razonamiento debe ser razonable
5. Debe poder utilizar un estándar para la representación de la información
6. Todo el conocimiento en la web podrá ser descrito usando este esquema
7. Estos estándares serán utilizados para la representación, consumo y compartición de información entre servicios y agentes

KIF (Knowledge Interchange Format) es un lenguaje desarrollado con el objetivo de intercambiar conocimiento entre los distintos sistemas basados en conocimiento y sistemas de inteligencia artificial. Este lenguaje fue definido dentro del proyecto Ontolingua, el cual proporciona un constructor ontologías cooperativo con la capacidad de exportar ontologías a diversos formalismos (Genesereth and Fikes, 1992).

RDF (Resource Description Framework) es un marco de trabajo para descripción de recursos recomendado por la W3C. RDF fue diseñado con el objetivo de representar metadatos en la web como derechos de autor, título, licencias de los documentos, así como acceder a listados de recursos compartidos. RDF también tiene la capacidad de representar información sobre cualquier recurso identificado (identified) en la Web.

El World Wide Consortium desarrolla este lenguaje en 6 bloques o especificaciones distintas:

1. RDF Primer<sup>1</sup>.
2. Resource Description Framework (RDF): Concepts and Abstract Syntax<sup>2</sup>.
3. RDF/XML Syntax Specification (Revised)<sup>3</sup>.
4. RDF Semantics<sup>4</sup>.
5. RDF Vocabulary Description Language 1.0: RDF Schema<sup>5</sup>.
6. RDF Test Cases<sup>6</sup>.

---

<sup>1</sup><http://www.w3.org/TR/rdf-primer/>

<sup>2</sup><http://www.w3.org/TR/rdf-concepts/>

<sup>3</sup><http://www.w3.org/TR/rdf-syntax-grammar/>

<sup>4</sup><http://www.w3.org/TR/rdf-mt/>

<sup>5</sup><http://www.w3.org/TR/rdf-schema/>

<sup>6</sup><http://www.w3.org/TR/rdf-testcases/>

El RDF es un estándar para descripciones simples. Mientras que el lenguaje XML sirve para la sintaxis, RDF se utiliza para representaciones semánticas debido a que tiene una serie de reglas para proveer información descriptiva.

DAML (DARPA Agent Markup Language) es un lenguaje de marcado de agente de la Agencia de Defensa de Estados Unidos de Investigación Avanzada de Proyectos (DARPA) que se basa en el lenguaje de marcado extensible (XML). DAML se compone de dos lenguajes: 1) lenguaje de ontología y 2) lenguaje para representar restricciones y reglas de inferencia. DAML está diseñado para tener una mayor capacidad que el lenguaje XML para la descripción de los objetos y sus relaciones, así como para expresar la semántica, y para crear un mayor nivel de interoperabilidad entre los sitios web.

OWL (Web Ontology Language) o Lenguaje de Ontologías para la Web, el cual es una recomendación del W3C (<http://www.w3.org/TR/owl-ref/>). OWL es un lenguaje de etiquetado semántico para publicar y compartir ontologías en la Web. OWL tiene la capacidad de representar ontologías de forma explícita, es decir, permite definir el significado de términos en *vocabularios* y sus *relaciones*. OWL surge como una revisión al lenguaje DAML-OIL y es considerado una extensión del lenguaje RDF, OWL emplea las tripletas de RDF y es más poderoso y expersivo que RDF y DAML-OIL.

En conclusión, las tecnologías de la Web Semántica permiten crear almacenes de datos en la Web, construir vocabularios, y escribir reglas para el manejo de datos<sup>7</sup>. Los lenguajes ontológicos son medios para expresar ontologías de forma comprensible por las máquinas. Algunos de ellos han emergido en los últimos años en paralelo a la idea de Web Semántica, de forma que están orientados para tal tecnología.

#### 2.2.1.4 Tecnologías semánticas dentro del reconocimiento de objetos

Como ya se mencionó, uno de los aportes más importantes de esta tesis la generación de escenas enriquecidas con datos con un sentido lógico, semántico y físico por medio de un algoritmo de reconocimiento de objetos, uno de los posibles usos es el facilitar el proceso de inventario de las instalaciones de fábricas. En este sentido, las obras más relevantes dentro del ámbito de las tecnologías semánticas aplicadas al reconocimiento 3D son descritas a continuación.

Un enfoque de la detección basada en el conocimiento de los objetos 3D utilizando el lenguaje de ontologías OWL llamada WiDOP fue presentado por Hmida et al. (2012). WiDOP utiliza el lenguaje VRML para definir una ontología de una escena indexada creada por un conjunto de nubes de puntos 3D. WiDOP tiene la capacidad de detectar objetos pero no los reconoce. A su vez, un modelo de visión por computador fue presentado por Gómez-Romero et al. (2011), el cual tiene el objetivo de construir un modelo

---

<sup>7</sup><http://www.w3.org/standards/semanticweb/>

simbólico de una escena de vídeo mediante la integración de datos de seguimiento e información contextual. Este modelo utiliza ontologías formales y usa procedimientos de razonamiento con 2 objetivos: 1) obtener una interpretación de alto nivel del escenario y 2) proporcionar información al procedimiento de seguimiento de bajo nivel para mejorar su precisión y rendimiento. Dentro del modelo se realiza razonamiento deductivo y abductivo para llevar a cabo la interpretación escena y la mejora de seguimiento. Tanto WiDOP como el modelo de Gomez-Romero localizan objetos dentro de las escenas 3D y 2D respectivamente, en contraste con la presente tesis, la cual tiene el objetivo de reconocer y clasificar cada elemento, además de calcular sus propiedades físicas y espaciales reales con el fin de enriquecer la escena escaneada.

Por otra parte, un algoritmo de retroalimentación basado en técnicas de extracción de características supervisadas fue presentado por Leifman et al. (2005). Este algoritmo tiene un motor de búsqueda Web que utiliza retroalimentación relevante para recuperar objetos semánticamente similares. Además, los autores presentan una anotación a través de etiquetas en modelos predefinidos en una base de datos. A su vez, Díaz-Vilariño et al. (2013) presentan una metodología para la generación automática de un Modelado de información de construcción (*BIM, Building Information Model*), el cual incluye superficies de sombreado que afectan a las necesidades de eficiencia energética e iluminación de edificios. Esta metodología describe un proceso de segmentación, el cual se lleva a cabo a través de un Algoritmo 3D de región de crecimiento, a través del cual se identifican las diferentes paredes y otros elementos del edificio. Dentro del sentido semántico se utilizan etiquetas de marcado para cada elemento, de modo que a cada región se le asigna una etiqueta de “Wall”, “Ceiling”, “Floor” y “Shade”. En consecuencia, el BIM resultante se puede someter a un análisis solar en el software de análisis de energía.

En este mismo sentido, Sengupta et al. (2013) propone un algoritmo que genera una reconstrucción 3D densa con etiquetados semánticos asociados. El objetivo de estas reconstrucciones 3D es apoyar la navegación y la localización dentro de una escena reconociendo su entorno para identificar e interactuar con objetos de interés. Todo esto a partir de un modelo semántico 3D, las entradas del algoritmo son pares de imágenes estéreo de nivel de la calle adquiridos a partir de una cámara montada en un vehículo en movimiento. Se calcula la profundidad de los mapas a partir de las imágenes y las posturas de dicha cámara, produciendo de esta manera una representación en 3D volumétrica de la escena. Las imágenes a nivel de calle se etiquetan de forma automática utilizando un marco de Campo Condicional Aleatorio (CRF, Conditional Random Field) por medio de la explotación de las imágenes estéreo adquiridas.

Tanto Díaz-Vilariño et al. (2013); Leifman et al. (2005), como Sengupta et al. (2013) utilizan un sistema de etiquetado simple, en cambio, la presente tesis ofrece un sentido semántico a los objetos reconocidos, permitiendo una descripción semántica más amplia de los objetos.

Otro ejemplo es el propuesto por León et al. (2016), en donde se presenta una firma

para la descripción semántica de formas 3D. A partir de una malla 3D segmentada y etiquetada, los autores utilizan distancias geodésicas a las diferentes etiquetas semánticas, codificando tanto información semántica como la geométrica, de acuerdo a la idea básica: dado un vértice, éste se caracteriza de acuerdo a sus distancias geodésicas a cada parte semántica del objeto. Aunque este trabajo se aproxima a la idea general de la presente tesis, existen algunas diferencias como el uso de la distancia geodésica por parte de Léon et al. (2016) para la caracterización de los elementos, en contraste con el razonamiento semántico presentado por esta tesis con el mismo propósito.

Finalmente, un conjunto de datos llamado IAIR-carped fue introducido por Y. Wu et al. (2012), los autores lo describen como el primer conjunto de datos para el reconocimiento de objetos de grano fino, se trata de una serie de capas de etiquetas semánticas anotadas por personas. Este conjunto de datos trabaja con escenas de calles de una ciudad con dos categorías: coches y peatones. Con este estudio los autores tratan de demostrar cómo las personas tienden a interpretar objetos mediante una categorización de grano fino, siendo esta más minuciosa en los detalles. Aunque IAIR-carped presenta una gran contribución a la clasificación de los peatones y coches, en el sentido semántico, sólo hay etiquetas asociadas a los elementos. En contraste la presente tesis tiene la capacidad de vincular sus elementos semánticamente definidas con otras fuentes de información basado en ontologías para enriquecer la escena.

### 2.2.2 Conclusiones del estado del arte

El reconocimiento es el proceso de identificación de objetos existentes en la imagen o escena, de acuerdo con la literatura consultada, el reconocimiento de objetos es el mayor reto de la visión por computador. Como se mencionó, las investigaciones desarrolladas en los últimos años reportan que en esta tarea no se ha logrado alcanzar las capacidades similares a las de un niño de 2 años (Granholm, 2015; Márton and Türker, 2013; Prince, 2012; Szeliski, 2010). Existen 4 tipos de reconocimiento de acuerdo al nivel de abstracción del objeto a reconocer (Szeliski, 2010): detección de objetos, reconocimiento de instancias, reconocimiento de categorías, comprensión de contexto y de escena. De acuerdo a esta clasificación, la tesis presentada está enfocada al reconocimiento de contexto y escena, debido a que se centra en el reconocimiento de objetos de acuerdo a su categoría geométrica pero bajo un contexto cerrado, específicamente instalaciones industriales.

Para desarrollar la tarea de reconocer un objeto dentro de una imagen o escena es necesario crear descriptores. Un descriptor visual es la descripción detallada de un objeto considerado en el proceso de reconocimiento de objetos (J. Wu and Rehg, 2011). Los descriptores pueden ser globales o locales. Los descriptores globales resumen el contenido de toda la imagen o escena en un único vector o matriz de características, mientras que los descriptores locales actúan sobre regiones de interés, previamente calculadas o identificadas, construyendo un vector de características de esa región.

También se mencionaron los subprocesos que componen al Proceso General de Re-



conocimiento de Objetos en una nube de puntos 3D: Captura, Pre-proceso, Definición de descriptores, Clasificación, Verificación e Interface, dicho proceso ha sido ampliamente estudiado en múltiples investigaciones como As Ari et al. (2014); Feng et al. (2015); Ivorra et al. (2015); Jensfelt and Kragic (2006), los cuales han enfocado sus trabajos en los subprocesos definición de descriptores y clasificación aplicados a enfoques como guías en trayectorias de robots móviles, reconocimiento de escenas en interiores controlados, en este sentido y de acuerdo con los objetivos de la tesis, esta investigación está centrada en los dos subprocesos más relevantes del reconocimiento de objetos: extracción de descriptores y clasificación, agregando dos supprocesos adicionales: pre-proceso y verificación.

En este estado del arte fueron presentados también diversos enfoques para la resolución del problema del reconocimiento de objetos 3D (Tangelder and Velkamp, 2008), los más relevantes para esta investigación son el enfoque basado en grafos topológicos y el enfoque basado en geometría. El enfoque basado en grafos topológicos distingue y reconoce la estructura, forma, y geometría de los objetos, al evaluar la estructura y forma de los objetos, no reconoce características detalladas de las superficies y falla en la clasificación cuando los objetos presentados contienen oclusiones parciales. Por otra parte, el enfoque basado en geometría soporta escenas desordenadas y oclusiones parciales y no utiliza descriptores basados en la apariencia, pero se limita al reconocimiento de formas simples y es sensible a ruidos y a deformaciones flexibles. Considerando las deficiencias de estos modelos, la presente investigación propone el uso de un algoritmo de reconocimiento de objetos en una nube de puntos 3D, enfocándose en optimizar los subprocesos de pre-procesamiento de la nube de puntos fuente, definición y categorización de descriptores para clasificar objetos en primitivas y verificación.

Por su parte, el análisis de la tecnología semántica permitió conocer el estado actual de la tecnología resaltando la importancia que tiene actualmente el uso de ontologías es sistemas basados en conocimiento. En relación a esta tesis doctoral la semántica ayudará a sustentar y completar el proceso de clasificación de objetos en una nube de puntos 3D. Al representar semánticamente la información de instalaciones industriales.

A partir de las limitaciones identificadas en el estado del arte, en el capítulo 3 presenta las hipótesis de la presente tesis, así como un conjunto de evaluaciones definidas para las mismas.



### 3.1 Planteamiento de las hipótesis a resolver

La problemática abordada en esta investigación para la creación de un algoritmo de reconocimiento y acotación semántica de primitivas en una nube de puntos 3D que permita enriquecer maquetas digitales, no tiene una solución trivial ni sencilla. El mayor reto se enfocará principalmente en la definición de los mecanismos que permitan el reconocimiento de primitivas geométricas y la creación de un modelo semántico, capaces de clasificar de una forma eficiente objetos definidos en una nube de puntos 3D y describirlos lógica, física y semánticamente dentro de un entorno industrial.

Como se ha comentado en la sección del estado del arte, existen diferentes investigaciones que se han ido desarrollando a lo largo de los años para dar solución al problema del reconocimiento de objetos 3D, los cuales fueron aplicados a diferentes ámbitos. Sin embargo estos trabajos presentan dificultades como sensibilidad a ruidos y a deformaciones flexibles o fallos en el reconocimiento de características detalladas de las superficies y en la clasificación de objetos con oclusiones parciales, lo que se deriva en la segmentación múltiple de un mismo objeto, además de carecer en su totalidad de información relevante que enriquezca a la nube de puntos más allá de las características geométricas. A diferencia de estos, en esta tesis se pretende la generación de un algoritmo inteligente basado en el análisis de descriptores visuales y la aplicación de tecnología semántica que optimice el reconocimiento y acotación semántica de primitivas enriqueciendo escenas reales a partir de la descripción de esos objetos que la componen con información lógica, física y semántica.

La información física se refiere al cálculo de características geométricas de cada objeto como alto, ancho, perímetro, radio, por mencionar algunas, Mientras que la información

semántica es obtenida mediante la asignación de cada objeto con su elemento correspondiente dentro de la ontología propuesta. Por último la información física se obtiene por medio del cálculo de la relaciones espaciales de los objetos dentro del plano cartesiano.

## 3.2 Hipótesis

Considerando las pretensiones descritas previamente para esta tesis, se procede a la descripción de la hipótesis a verificar:

- **Hipótesis H1:** Es posible mejorar el reconocimiento de objetos a partir de la acotación semántica de primitivas que describan elementos comunes dentro de una escena industrial, específicamente cilindros, toros y esferas pertenecientes a una nube de puntos 3D considerando la combinación de descriptores de características geométricas y el uso de reglas semánticas.
- **Hipótesis H2:** Es posible enriquecer una nube de puntos detallando la definición de sus objetos de acuerdo al cálculo de sus características a partir de descriptores visuales.
- **Hipótesis H3:** La información proporcionada por este algoritmo puede aplicarse al reconocimiento de un entorno real.

Tamayo (2004) menciona que uno de los propósitos de las hipótesis es servir de directrices en una investigación, por lo que, comprobar las hipótesis planteadas resulta imperativo para esta tesis. En la siguiente sección se plantea el desarrollo de los artefactos tecnológicos necesarios para una correcta validación y evaluación de las hipótesis.

## 3.3 Validación de las hipótesis

El método científico está basado en la experimentación, es decir, que consiste en usar los datos obtenidos para formular una suposición o hipótesis sobre un modelo en particular para posteriormente realizar experimentos adicionales con el fin de afirmar o rechazar la suposición (Moret and Shapiro, 2001).



Fig. 3.3.1: Etapas del estudio experimental para el reconocimiento de primitivas en una nube de puntos 3D

En un estudio experimental el proceso de diseño de los experimentos es de gran importancia, debido a que cada experimento sirve como una herramienta de descubrimiento o un

medio para contestar preguntas específicas (Moret and Shapiro, 2001). El proceso general de estudio experimental inicia por formular una hipótesis, seguido de la descripción del medio o recursos por los cuales se adquieren los datos para probar las hipótesis, después se asegura la reproducción del experimento y finalmente la diseminación de la importancia de los resultados. En este sentido, en esta sección se presenta el diseño del estudio experimental personalizado para la evaluación de esta tesis, la cual nos ayudará a comprobar las hipótesis H1, H2 y H3 propuestas. El diseño de este estudio experimental comprende primordialmente cuatro etapas, las cuales se muestran en la Figura 3.3.1 y se describen brevemente a continuación.

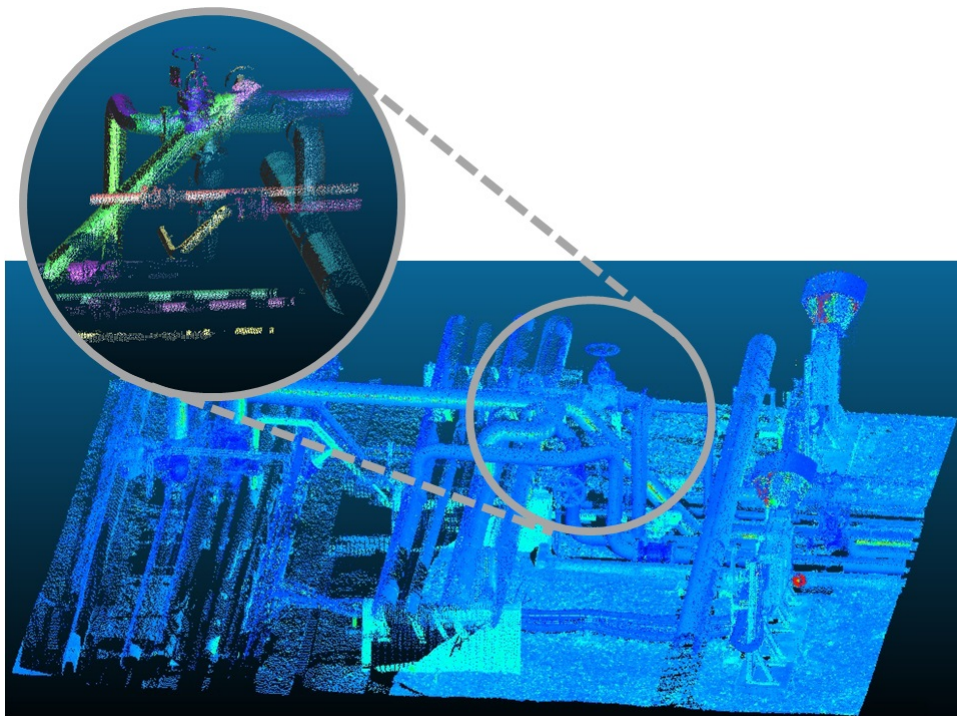


Fig. 3.3.2: Nube de puntos 3D

- **Análisis de datos fuente:** Representa la primera etapa de este estudio y se centra en el análisis de los datos fuente, dicho análisis incluye el identificar el formato de la nube de puntos proporcionada por el escáner, si este se rige por algún estándar, tiene algún orden en específico, por nombrar algunas. Cabe mencionar que para el desarrollo de esta etapa se dispone de datos reales suficientes, la figura 3 muestra un ejemplo de estos datos, una la nube de puntos robusta compuesta por 1922097 puntos. Esta etapa es el punto de partida para llevar acabo la verificación de cada una de las hipótesis planteadas en esta tesis.

- **Algoritmo Inteligente:** Es necesario el desarrollo de un algoritmo basado en descriptores visuales y tecnología semántica que optimice el reconocimiento de primitivas combinando el uso de descriptores de características geométricas y reglas semánticas, y además enriquezca la descripción de esos objetos incluyendo información **lógica, física y semántica** de una nube de puntos en un entorno industrial. La definición del algoritmo inteligente contribuirá a la comprobación de las hipótesis H1 y H3. Esto se debe a la clasificación correcta de los objetos dentro de la nube de puntos en un ambiente industrial.
- **Verificación:** La etapa final es donde el conjunto de objetos dentro de la nube de puntos fueron clasificados de acuerdo a sus características **lógicas, semánticas y físicas**. Para verificar esta clasificación se pueden utilizar algunos de los siguientes criterios:
  - Cálculo del número de falsos positivos FP, falsos negativos FN, verdaderos positivos TP y verdaderos negativos TN.
  - Curvas de confiabilidad generadas a partir de la medida de confiabilidad
  - Curva de Medida-F generada a partir de la sensibilidad de la clasificación.

Esto se traduce en la comprobación de la hipótesis H3.

Como se menciona en este capítulo, la tesis abordará los cuatro subprocesos mas relevantes del proceso de reconocimiento de objetos en nubes de puntos 3D: pre-proceso, extracción de descriptores, clasificación y verificación. Por tal motivo, es importante proporcionar su descripción más a detalle, lo cual ocurre en el capítulo siguiente.

## CAPÍTULO 4

---

### Algoritmo inteligente

---

Tal como se presentó en el estado del arte, existen diversos esfuerzos para crear maquetas digitales de escenas en diferentes contextos. Dichas maquetas presentan información acerca de la superficie de los objetos, dimensiones generales o incluso hasta etiquetas semánticas específicas. Sin embargo para los objetivos de esta tesis, esta información no es suficiente, se requiere que los objetos sea descritos semánticamente a detalle y las relaciones que existen entre ellos. El desafío entonces consiste en describir el entorno que las personas percibimos en una escena virtual, de tal manera que sus propiedades físicas, lógicas y semánticas sean detalladas. En este sentido, esta tesis propone un algoritmo de ingeniería inversa, capaz de reconocer y acotar semánticamente primitivas geométricas, que describan elementos comunes dentro de una escena real, en una nube de puntos captados vía un escáner. Esto permitirá la generación de una maqueta digital en donde los objetos serán descritos con un alto grado de detalle, proporcionando no solo la forma, sino también características geométricas como alto, largo, diámetro; sus relaciones espaciales con otros objetos e información semántica relacionada.

### 4.1 Descripción general

De acuerdo al capítulo 3, el desarrollo de un algoritmo inteligente para el reconocimiento y acotación semántica de primitivas en una nube de puntos 3D ayudará a comprobar las hipótesis H1, H2 y H3 propuestas. En este capítulo se describirá de manera general el algoritmo propuesto, mientras que en el Capítulo 5 se describirá una implementación del mismo bajo un contexto industrial.

El algoritmo propuesto está compuesto por los cuatro subprocesos mas relevantes del

proceso de reconocimiento de objetos (ver Figura 2.1.1) en nubes de puntos 3D: pre-proceso, definición de descriptores, clasificación y verificación, los cuales se describirán a lo largo del presente capítulo.

Dentro del contexto computacional, un algoritmo es definido como un procedimiento computacional que coge algunos valores o un conjunto de valores, como *entrada*, los procesa y genera algunos valores o conjunto de valores como *salida*. Por lo que un algoritmo es una secuencia de pasos computacionales, los cuales transforman las *entradas* a *salidas* (Cormen, 2009). En lo que se refiere a esta tesis, la *entrada* entonces corresponde a la nube de puntos de una escena real, mientras que la *salida* está conformada por un conjunto de datos lógicos, físicos y semánticos que describen los objetos que componen dicha escena.

En este sentido, el algoritmo propuesto se ha dividido en dos procesos: Reconocimiento de primitivas y enriquecimiento semántico, tal y como muestra la Figura 4.1.1. Cada uno de los procesos tiene objetivos diferentes, el primero tiene el objetivo de clasificar la nube de puntos en segmentos con etiquetas simples, mientras que el segundo proceso pretende proveer a la maqueta digital de información enriquecida acerca de los segmentos reconocidos. Estos dos procesos se complementan de manera que su desarrollo e implementación permiten la validación de las Hipótesis H1, H2 y H3.

Tal y como lo muestra la Figura 4.1.2, el algoritmo entonces inicia con la recepción de una nube de puntos de una escena real mostrada desde un láser escáner, después toca el turno del proceso nombrado *Reconocimiento de Primitivas*, el cual analiza la nube de puntos y la segmenta de acuerdo a sus características geométricas, generando una clasificación simple de los elementos, es decir, por asignación de etiquetas de acuerdo a la clasificación, pero aún sin un sentido semántico formal. Y crea una lista de segmentos etiquetados. Dicha lista pasa ahora al proceso de *Enriquecimiento Semántico*, en donde se calculan propiedades específicas de acuerdo al tipo de elemento etiquetado, como por ejemplo largo, ancho, radio, radio mayor, radio menor, diámetro, por mencionar algunos; y también se calcula las relaciones espaciales de los segmentos por medio de reglas semánticas. El alcance de estas reglas semánticas genera que algunos segmentos se unan con otros, solucionando así problemas de redundancia y de múltiple segmentación de un mismo objeto. Debido a estas agrupaciones y a las validaciones semántica y por expertos, surge la necesidad de volver a calcular las características geométricas de los nuevos elementos formados y eliminar los redundantes.

La acotación semántica del algoritmo se genera a partir de la definición de un modelo ontológico semántico, el cual contendrá la información calculada y asociada a los elementos que componen la maqueta digital.

El resultado será una representación lógica, física y semántica de los objetos que componen la escena virtual. Cabe señalar que este algoritmo es una referencia general y que cada uno de los procesos, subprocesos y modelo semántico debe ser implementado de acuerdo al enfoque de estudio. Los dos procesos implicados y el modelo ontológico propuesto son descritos a continuación con el objetivo de aclarar los detalles del algoritmo propuesto en



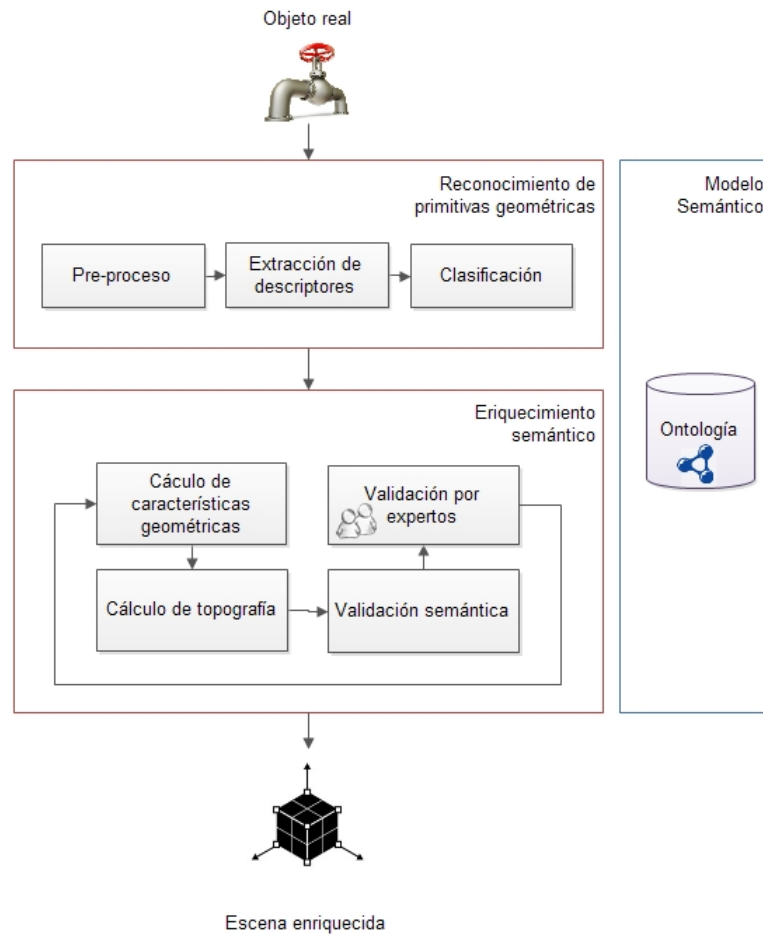


Fig. 4.1.1: Algoritmo para el enriquecimiento de de escenas 3D

la presente tesis.

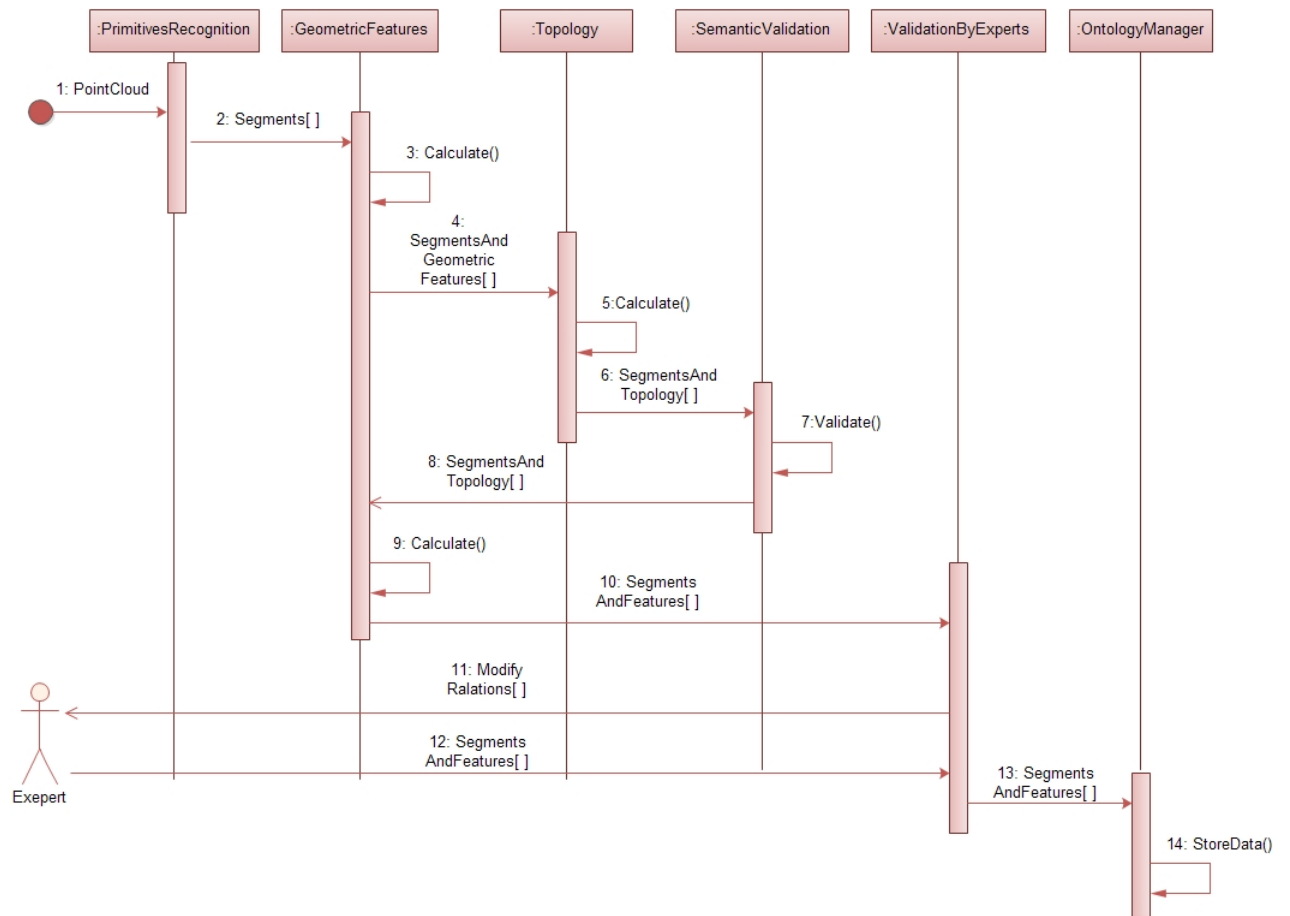


Fig. 4.1.2: Diagrama de secuencia del algoritmo propuesto

## 4.2 Proceso de reconocimiento de primitivas

El proceso de reconocimiento de primitivas tiene el objetivo general dentro de esta tesis de responder a la pregunta *¿Existe algún objeto conocido dentro de la escena escaneada?*. Para resolver este cuestionamiento se llevan a cabo tres tareas, todos ellos con el objetivo específico conjunto de segmentar una nube de puntos de acuerdo a sus características geométricas. La *entrada* será entonces, una **nube de puntos** y la *salida* una **lista de segmentos clasificados** de acuerdo a sus características geométricas.

Este proceso se divide en 3 tareas:

1. Pre-proceso
2. Definición de descriptores
3. Clasificación

### 4.2.1 Pre-proceso

El pre-proceso tiene como objetivo reducir la información de entrada mejorando la calidad de la escena original (Gonzalez, 2009). En esta tarea se elimina la información irrelevante, es decir el ruido, así como la redundante que se encuentre en la escena; además la nube de puntos es ordenada en una estructura jerárquica, específicamente un árbol *kd* y finalmente se calculan datos que son requeridos para la definición de descriptores como las normales de los puntos.

#### 4.2.1.1 Árbol *kd*

Un árbol *kd*, o árbol *k*-dimensional, es una estructura de datos jerárquica utilizada para la organización de un conjunto finito de puntos en un espacio euclídeo de *k* dimensiones (Moore and Hall, 1990). Es un árbol de búsqueda binaria en el cual todos los nodos, desde el raíz hasta los nodos hojas almacenan un punto. Árboles *kd* son muy útiles para el almacenamiento y las búsquedas de vecinos más cercanos. De acuerdo a las características de la nube de puntos utilizada en la presente tesis, se utilizan árboles *kd* tridimensionales. Técnicamente, la letra *k* se refiere al número de dimensiones. Un árbol *kd* tridimensional podría ser llamado un árbol 3d. Sin embargo se suele emplear la expresión "árbol *kd* tridimensional".

Un árbol *kd* para un conjunto de elementos  $P = (p_1, p_2, \dots, p_N)$  es un árbol binario que cumple con las siguientes características:

- Cada nodo perteneciente a  $P$  contiene una llave multi-dimensional que describe la coordenada del punto; un discriminante y dos punteros, los cuales pueden ser nulos o apuntar al siguiente nodo hijo dentro del árbol.

- Cada nodo  $p_i$  que pertenece a  $P$  cumple con las afirmaciones:
  - Todos los elementos del sub-árbol izquierdo de  $p_i$  tienen un valor menor para el discriminante del nodo que el de  $p_i$
  - Todos los elementos del sub-árbol derecho de  $p_i$  tienen un valor mayor o igual para el discriminante del nodo que el de  $p_i$

El árbol  $kd$  es una estructura balanceada donde cada nodo hoja está a la misma distancia de la raíz.

La Figura 4.2.2 muestra a) un árbol  $kd$  de dos dimensiones, con los seis elementos siguientes: (6,3), (2,5), (3,8), (8,9), (7,2), (2,3) y b) un plano euclidiano dividido por el árbol generado.

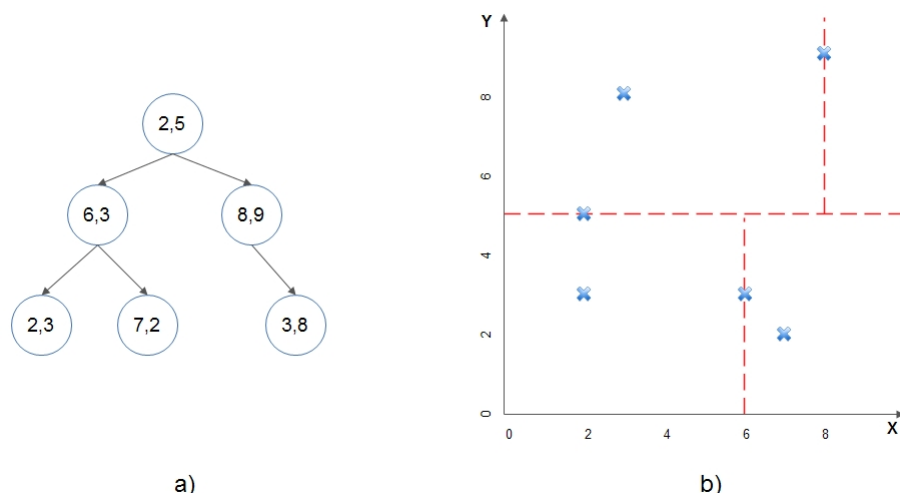


Fig. 4.2.1: Ejemplo de un árbol  $kd$  de dos dimensiones a) muestra un árbol  $kd$  de seis elementos, donde el nodo raíz divide a través del plano Y; b) muestra un plano  $x,y$  dividido por el árbol

En este sentido, se utilizó la biblioteca ANN<sup>1</sup> para generar la estructura de árbol  $kd$ . ANN es una biblioteca creada por David M. Monte y Sunil Arya de la Universidad de Maryland y de la Universidad de Ciencia y Tecnología de Hong Kong respectivamente. La biblioteca ANN (*Approximate Nearest Neighbor*) apoya las búsquedas exactas y aproximadas del vecino más cercano en espacios de diferentes dimensiones.

En esta tesis ANN es utilizada como apoyo para la búsqueda del vecino más cercano en las tareas de Normalización y Extracción de descriptores.

<sup>1</sup><http://www.cs.umd.edu/~mount/ANN/>

El problema del vecino más cercano es siempre aplicado a un conjunto finito de datos, en este caso la nube de puntos escaneada, estos puntos son procesados en una estructura de datos *kdtree* soportada por la propia librería ANN. La distancia entre dos puntos se puede definir de muchas maneras, ANN asume que las distancias son medidas utilizando cualquier clase de funciones de distancia llamadas Métricas Minkowski. Estos incluyen la conocida distancia euclidiana y la distancia Manhattan. En el caso de esta tesis se seleccionó la distancia Euclidiana.

#### 4.2.1.2 Normalización

Tomando el término normalización en su sentido más etimológico puede entenderse como “ajustar algo a una norma”, es decir, procesar la información origen a una norma de manera que dos conjuntos diferentes de datos sean comparables.

Desde el punto de vista matemático la normalización es la determinación de un vector unitario de la misma dirección y sentido que uno dado, pero distinto del vector cero. Si el vector  $v$  es distinto de  $0$ , entonces su normalización se efectúa dividiéndolo entre su módulo, tal y como se expresa en 4.2.1:

$$n = \frac{v}{||v||} \quad (4.2.1)$$

Al proceso de obtener un vector unitario asociado a un vector  $v$  se le llama *normalización del vector*, por lo cual es común referirse a un vector unitario como *vector normalizado*.

La casi nula información obtenida de los ficheros fuente crea la necesidad de calcular distintas medidas invariantes, el más relevante es la normal. El proceso de reconocimiento de primitivas en nubes de puntos en tres dimensiones precisan de la normalización previa con el objetivo de convertir la escena 3D original en una escena 3D canónica invariante a la traslación, a la rotación y a la escala (Tangelder and Velkamp, 2008; Yang et al., 2004). Este proceso es fundamental para poder comparar los objetos de la nube de puntos con los descriptores y para que la cuantificación de las características extraídas posteriormente se corresponda con las medidas reales de los objetos.

Por todo lo anterior, en el contexto de la tesis *normalizar* significa procesar la escena de modo que no importe como se encuentra su contenido, siempre éste se pueda mapear a una estructura común la cual pueda describir de forma invariante a la transformación afín.

Se puede decir entonces que hay diferentes tipos de normalización. Uno de ellos se realiza respecto del centro de masas de la escena, es decir, respecto al centro de la superficie de puntos que define la imagen. Para normalizar respecto de la escala, la distancia entre cada punto y el centro de masas debe ser escalada por una constante. Con respecto a la

traslación, el centro de masas debe ser trasladado al origen del sistema de coordenadas (Tangelder and Veltkamp, 2008).

En aplicaciones de gráficos por ordenador, la propiedad más importante de una superficie geométrica es la *normal de la superficie*, la cual define la orientación de cada punto de la superficie en un sistema de coordenadas. La normal a una superficie en un punto  $P$  es un vector perpendicular al plano tangente a la superficie en  $P$ . El concepto de normalidad se generaliza a la ortogonalidad, es decir, determina la orientación de una superficie hacia una fuente de luz para el sombreado plano, o la orientación del punto dado para imitar una superficie curva (Blinn, 1978).

La normal de la superficie puede calcularse de dos formas, las cuales consideran aspectos diferentes:

- Considerando la superficie subyacente del punto, utilizando técnicas de mallado de superficie, y luego calcular las normales a la superficie de la malla. Por lo que, la estimación de la normal en un vértice, se obtiene a partir del promedio de las normales a cada una de las caras o triángulos adyacentes (Woo et al., 2002)
- Utilizando aproximaciones para inferir las normales de la superficie de la nube de puntos de datos directamente (Hoppe et al., 1992)

En esta tesis nos centramos en la última, es decir, estimar directamente las normales determinando un vecindario alrededor de cada punto (para aproximar la superficie localmente).

Primero se ajusta un plano mediante mínimos cuadrados con  $k$  puntos alrededor del punto analizado. Los mínimos cuadrados es un procedimiento que permite encontrar un plano que tenga una distancia mínima a todos los puntos sobre los que se está ajustando.

El plano se puede escribir en términos de un vector normal como en 4.2.2, donde el vector  $\vec{n} = (A, B, C)$  en un **vector normal** al plano, es decir, **perpendicular al plano**.

$$\vec{p}_{i,j} \cdot \hat{n} = d \quad (4.2.2)$$

Si  $P(x_0, y_0, z_0)$  es un punto del plano, y el vector  $\vec{PX} = (x - x_0, y - y_0, z - z_0)$  es perpendicular al vector  $\vec{n}$  y por lo tanto en 4.2.3 el producto escalar es cero.

$$\vec{PX} \cdot \vec{n} = 0 \quad (4.2.3)$$

Pero debido a que los puntos a analizar no están exactamente dentro de un plano, en 4.2.2 se sustituye  $d$ .

$$\vec{PX} \cdot \vec{n} = d \quad (4.2.4)$$

Al aproximar los puntos se genera un error de mínimos cuadrados, el cual se expresa en 4.2.5.

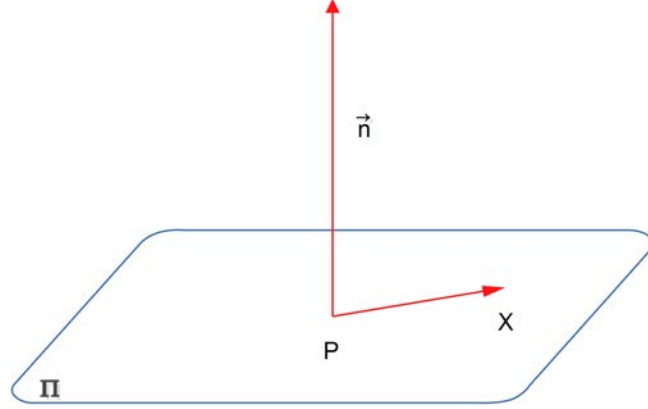


Fig. 4.2.2: Vector normal

$$\varepsilon = \sum_{k,l \in W_{i,j}} (\vec{P}_{k,l}^T \hat{n} - d)^2 \quad (4.2.5)$$

$$\varepsilon = \hat{n}^T \left( \underbrace{\sum_{k,l \in W_{i,j}} \vec{P}_{k,l} \vec{P}_{k,l}^T}_Q \right) \hat{n} - 2d \left( \underbrace{\sum_{k,l \in W_{i,j}} \vec{P}_{k,l}^T}_{\vec{q}^T} \right) \hat{n} + N^2 d^2$$

A continuación se coge el siguiente lagrangiano 4.2.6.

$$l(\hat{n}, d, \lambda) = \varepsilon + \lambda(1 - \hat{n}^T \hat{n}) \quad (4.2.6)$$

Derivando respecto a  $n$  y  $d$ , y substituyendo, se llega a 4.2.7.

$$Q(\hat{n} - \frac{\vec{q}\vec{q}^T \hat{n}}{N^2}) - \lambda = 0 \quad (4.2.7)$$

se factoriza  $n$  por la derecha en los dos primeros términos, por lo que se tiene 4.2.8.

$$(Q - \frac{\vec{q}\vec{q}^T}{N^2})n - \lambda n = 0 \quad (4.2.8)$$

o bien, pasando  $\lambda n$  a la derecha como se muestra en 4.2.9.

$$(Q - \frac{\vec{q}\vec{q}^T}{N^2})n = \lambda n \quad (4.2.9)$$

Se substituye con  $R$  y queda 4.2.10.

$$Rn = \lambda n \quad (4.2.10)$$

Donde

$$R = (Q - \frac{\vec{q}\vec{q}^T}{N^2}) \quad (4.2.11)$$

Entonces, en ese tipo de ecuaciones  $R$  es una matriz,  $n$  es un vector llamado autovector o vector propio y  $\lambda$  es un número llamado autovalor o valor propio. Por lo que, el problema de encontrar el vector normal se convierte en el problema de encontrar los vectores propios para la matriz  $R$ .

Esa matriz  $R$  se construye con  $Q$  y  $\vec{q}\vec{q}^T$ , estas matrices se forman a partir de los vectores que entre el punto se está analizando y los puntos k-vecinos. Dice ahí que el error coincide con el valor propio, entonces, como se pretende que el error sea mínimo, se busca el valor propio más pequeño de la matriz y el vector propio que tenga asociado es el vector normal que se busca.

La Figura 4.2.3 muestra un ejemplo de superficies normales estimados para un conjunto de puntos.

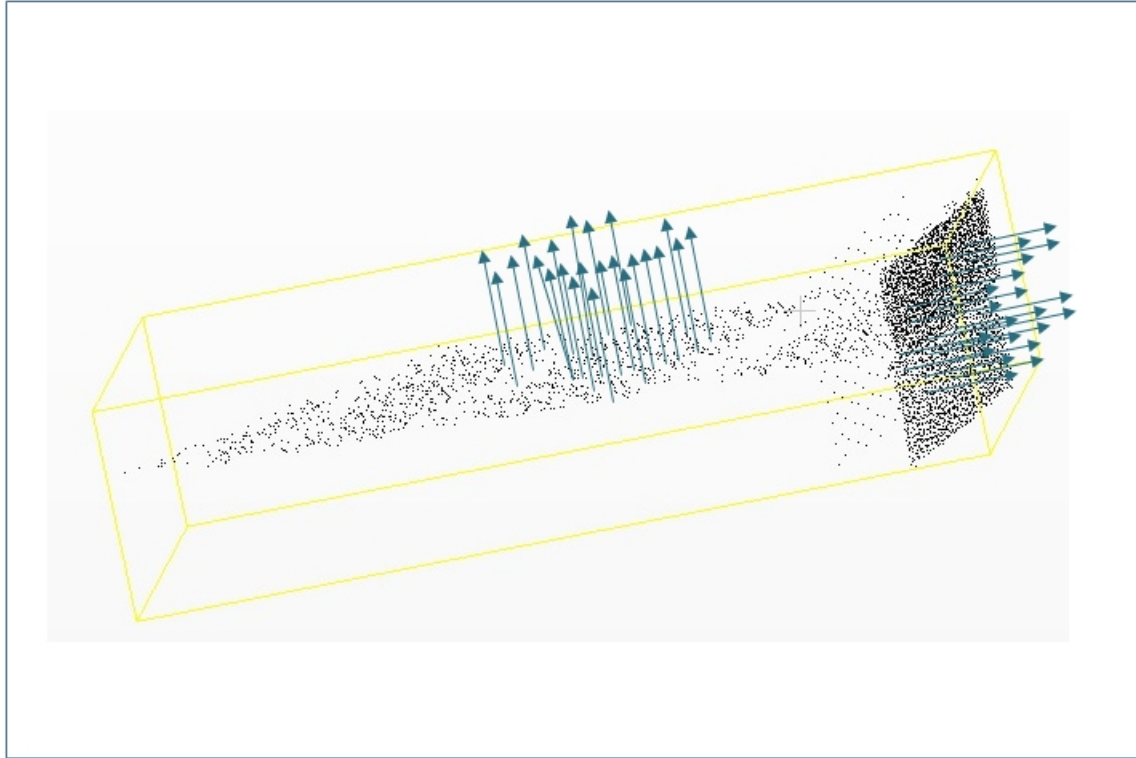


Fig. 4.2.3: Ejemplo de superficies normales estimados para un conjunto de puntos. figura adaptada de R. Rusu (2010)

Una vez concluido el pre-proceso donde se eliminó el ruido, el resultado es una nube



de puntos ordenada en una estructura jerárquica, un árbol kd, y además con normales de los puntos calculadas. A continuación se procede con la definición de descriptores.

#### 4.2.2 Definición de descriptores

La definición de descriptores tiene el objetivo de reducir al mínimo la cantidad de información obteniendo características específicas que permitan representar un fenómeno o problema y describirlo matemáticamente (Cabral et al., 2005). Esto puede ser posible mediante un modelo matemático, el cual define un conjunto de símbolos y relaciones matemáticas que representa, de alguna manera, el fenómeno en cuestión (Biembengut and Hein, 1999), en el caso aplicativo del presente algoritmo, se definen primitivas geométricas, las cuales serán posteriormente reconocidas en una nube de puntos.

En resumen, dentro del reconocimiento de primitivas geométricas, cada primitiva se identifica por medio de sus parámetros (definidos en este proceso), por tanto, el resultado de este paso será una lista de descriptores, es decir modelos matemáticos que definen primitivas geométricas. El tipo de primitivas a reconocer, así como los parámetros que los describen serán definidos de acuerdo a la implementación y el contexto del algoritmo.

Después, cada conjunto de datos es comparado con la definición de las primitivas predefinidas y se calculará el error de ajuste.

#### 4.2.3 Clasificación

La clasificación es el proceso donde se decide la categoría más probable a la que pertenece cada segmento (también llamado conjunto de puntos) considerando como discriminante sus propiedades y su error de ajuste (Attene, Falcidieno, et al., 2006; Attene and Patane, 2010; Diaz et al., 2004; Rabbani and Heuvel, 2005).

Por lo que, esta sección coge como entrada la nube de puntos ordenada mediante una estructura jerárquica de datos, las normales de los puntos y la definición de los descriptores de las primitivas a reconocer y va formando segmentos los cuales son discriminados de acuerdo a sus propiedades y error de ajuste.

Esta sección fue diseñada con un enfoque genérico de manera que tenga la capacidad de soportar diferentes APIs de algoritmos de reconocimiento que segmenten elementos. La implementación de este proceso de clasificación de segmentos dependerá del contexto en el que se desarrolle el algoritmo. Las necesidades a cumplir son:

1. Tener la capacidad de procesar y segmentar nubes de puntos.
2. Tener la capacidad de reconocer primitivas geométricas, específicamente cilindros, toros y esferas.
3. Calidad en el segmentado y clasificación de las primitivas geométricas.

En este paso, por tanto, se implementarán APIs de algoritmos de reconocimiento que clasifiquen segmentos de acuerdo a un contexto, de forma que el resultado es una lista de elementos segmentados y clasificados por su tipo más probable pertenecientes a la nube de puntos, la cual servirá de entrada para el siguiente proceso, es decir el enriquecimiento semántico.

### 4.3 Enriquecimiento semántico

El proceso de enriquecimiento semántico pretende ampliar el proceso de reconocimiento de objetos de tal forma que se permita generar datos enriquecidos a partir de una nube de puntos 3D. De esta manera este proceso pretende resolver la pregunta *¿Qué tipo de elementos tiene la escena escaneada?*.

Este proceso del algoritmo tiene como *entrada* una lista de segmentos clasificados a un tipo de primitiva de acuerdo a sus características geométricas, mientras que la *salida* está conformada por un conjunto de datos lógicos, físicos y semánticos que describen los objetos que componen la escena escaneada.

Tanto las primitivas geométricas a reconocer, como sus características son dependientes del dominio en el que se implementará el algoritmo. En el apartado 5 se desarrolla un caso de uso concreto, en donde se puede ampliar la visión del Enriquecimiento semántico.

Este proceso se divide en 2 tareas:

- Cálculo de características geométricas
- Cálculo de la topología

El enriquecimiento semántico inicia con la tarea *Cálculo de características geométricas*, donde se analizan las propiedades que se calcularán por cada primitiva, esto dependerá del contexto en el que se implemente el algoritmo. Una vez definidas las propiedades geométricas específicas a calcular, se procede con la extracción de las mismas de acuerdo con la primitiva asociada por el proceso anterior.

La siguiente tarea es el *Cálculo de la topología*, es decir las relaciones espaciales entre los objetos pertenecientes a la nube de puntos. Después de eso, toda la información calculada se almacena, considerando la definición de una ontología propuesta. El resultado final será una maqueta enriquecida.

De modo que el algoritmo tiene la capacidad de generar información relevante a partir de la nube de puntos original, es decir, información que permita describir los objetos pertenecientes a la escena escaneada de una manera lógica, física y semántica; lógicamente, mediante el cálculo de características geométricas tales como altura, ancho, perímetro, radio, por mencionar algunas; semánticamente haciendo coincidir cada objeto con su elemento correspondiente en la ontología propuesta, la cual se describirá más adelante; y en cuanto a su sentido físico mediante el cálculo de las relaciones espaciales y propiedades como la posición, el tamaño o el número de puntos.

### 4.3.1 Definición de la ontología

Dentro del algoritmo propuesto, el modelo semántico será definido por reglas semánticas que definirán la topología de los objetos (las cuales se abordarán en las siguientes secciones) y por una ontología formal que describa primitivas geométricas relacionadas entre si, que imiten lo más cercano posible el comportamiento de los objetos reales asociados a ellos. La Figura 4.3.1 muestra la ontología general propuesta, a partir de la cual se generarán especificaciones de primitivas de acuerdo al contexto en el que se implemente el presente algoritmo.

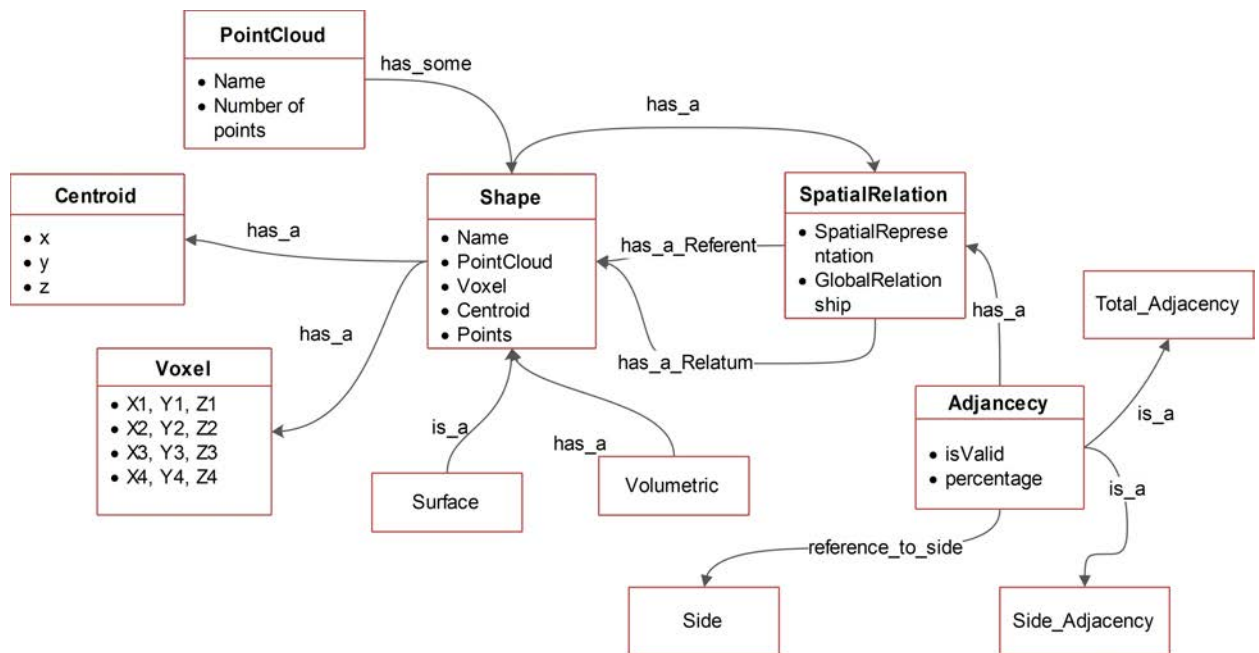


Fig. 4.3.1: Ontología general

El diseño y la definición de cada elemento de la ontología es descrito en el apartado 5.2.2.1, donde se complementa con elementos pertenecientes a la implementación de un caso de uso concreto.

D. Chen et al., 2014 definen la ontología como “datos que describen datos”. Una ontología puede ayudar a un usuario a aprender y a encontrar la información relevante en un contexto. En el caso del algoritmo, el modelo semántico permitirá consultar información acerca de la escena y hacer inferencias con sus datos.

La combinación de primitivas geométricas dentro de la ontología, permite representar figuras complejas presentes en el mundo real. La ontología propuesta define los objetos reconocidos por el algoritmo, los cuales pueden estar presentes en diferentes escenas, por ejemplo, la escena de interior de una fábrica. Además, tiene la posibilidad de enlazarse

con ontologías externas, que enriquecerán su significado semántico.

### 4.3.2 Cálculo de características geométricas

Esta tarea tiene como objetivo calcular propiedades geométricas específicas de cada elemento de acuerdo con el tipo de primitiva geométrica clasificada, tales como longitud, diámetro, altura y el radio. Dichas propiedades van poblando la ontología con su correspondiente objeto.

### 4.3.3 Cálculo de topología

Topología se define como un modelo matemático utilizado para representar la ubicación y las relaciones entre objetos reales. La topología puede representar la conectividad de una entidad a otra; por ejemplo, un borde tendrá relaciones topológicas a su forma y a sus nodos (L. Liu and Zsu, 2009).

Dentro del campo de las matemáticas, la topología investiga las características de la geometría que permanecen invariables bajo ciertas transformaciones, es decir asignaciones topológicas o homeomorfismo (Egenhofer and Sharma, 1993). Asignaciones topológicas son transformaciones elásticas que mantienen las relaciones de incidencia sin cambios topológicos, esto es, aunque existan cambios físicos de la forma, las propiedades topológicas son invariantes. La topología tiene dos grandes ramas: 1) la topología de conjunto y 2) la topología algebraica. Ambos aplican para el manejo de datos espaciales.

Para el cálculo de la topología de los elementos dentro de las escenas, el algoritmo calculará las **relaciones espaciales** y la **representación espacial** de los elementos, los cuales van poblando la ontología y creando relaciones semánticas entre los elementos existentes. El cálculo de las **relaciones espaciales** y la **representación espacial** es descrita a continuación.

#### 4.3.3.1 Relaciones espaciales

La relación direccional entre objetos especifica el rango posible de posiciones en las que un objeto en una descripción debe aparecer con respecto a la posición de otro objeto (C. Li et al., 2009). La posición o relación espacial de los objetos se puede describir en tres formas diferentes intrínsecas, relativos y absolutos (Levinson, 1996; Méndez et al., 2014; Moratz and Ragni, 2008; Moratz, Tenbrink, et al., 2003). Los cálculos de posición cualitativos son modelos computacionales que describen sistemas de referencia relativa de las proyecciones de relaciones, es decir, *Front*, *Back*, *Left*, *Right*, *Above* y *Below*. Las relaciones espaciales entre los objetos se describen de acuerdo a la posición del primer objeto relativo llamado *Refent*, hacia la orientación intrínseca de otro objeto denominado *Relatum*. En un sistema de referencia relativa, la posición relativa de un *Referent* a su *Relatum* se describe desde el punto de vista de la tercera posición llamado *Origen* (Hois et al., 2006). Es decir, la posición espacial relativa de un objeto en función del ángulo de visión de la que se observa

y su sensibilidad al ángulo de rotación de la figura. Por lo tanto, para esta tesis el *Origen* es igual a la posición del láser con respecto a la escena.

El algoritmo intenta imitar con reglas semánticas el comportamiento de los objetos reales que podemos encontrar en la nube de puntos. Para este caso, se crearon reglas semánticas para las relaciones espaciales de los objetos denominadas *SpatialRelationshipRules*.

#### 4.3.3.2 Representación espacial

Una topología general de objetos permite conocer las interacciones espaciales que tienen entre ellos (Egenhofer and Herring, 1990). Las relaciones topológicas de un objeto espacial se definen con respecto a las características y necesidades de cada modelo espacial. Por lo que, las relaciones topológicas principales consideradas para los casos exclusivos de objetos 3D sin agujeros incrustados son definidos como “disjoint, meet, equal, overlap, inside, contains, covers y covered by”. En la Figura 4.3.2 se muestra un ejemplo de cada relación.

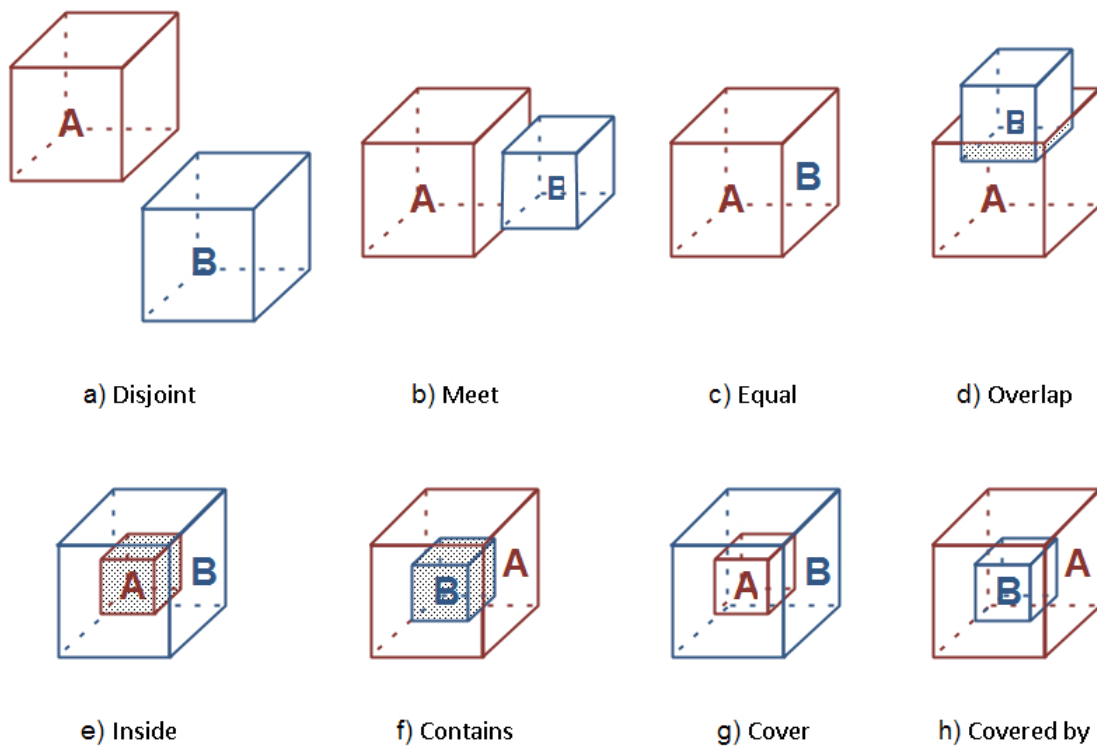


Fig. 4.3.2: Representación espacial de las formas A y B. Basado en C. Li et al. (2009); Zlatanova et al. (2004)

En un sentido algebraico, las primitivas fundamentales de una topología son: exterior, interior y frontera de un subconjunto  $A$  de un espacio topológico, el cual es representado como  $\bar{A}$  (exterior),  $A^\circ$  (interior) y  $\delta A$  (límite), respectivamente.

Todas las posibles relaciones binarias entre dos regiones  $\mathbf{A}$  y  $\mathbf{B}$  son derivadas del análisis de las cuatro intersecciones del límite y el interior de  $\mathbf{A}$  con el límite y el interior de  $\mathbf{B}$  (Egenhofer and Franzosa, 1991). Esto puede ser representado por la matriz de 2x2

$$I_4(A, B) = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \delta B \\ \delta A \cap B^\circ & \delta A \cap \delta B \end{pmatrix} \quad (4.3.1)$$

Considerando que estas 4 intersecciones pueden retornar vacíos  $\emptyset$  o no vacío  $\neg\emptyset$  y que son invariantes topológicos, se pueden detectar ocho relaciones topológicas binarias: *disjoint*, *contains*, *inside*, *equal*, *meet*, *covers*, *coveredBy* y *overlap* (Guptill and Morrison, 2013).

- **Disjoint.** Si las cuatro intersecciones entre todas las caras no existen, entonces las dos áreas están separadas.

$$R_{disjoint}(A.B) \Leftrightarrow I(A.B) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (4.3.2)$$

La Figura 4.3.2 (a) muestra la relación Disjoint.

- **Meet.** Si la intersección entre los límites de los objetos existe y las otras tres intersecciones no se cumplen, entonces estos objetos se tocan.

$$R_{meet}(A.B) \Leftrightarrow I(A.B) = \begin{pmatrix} \neg 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (4.3.3)$$

La Figura 4.3.2 (b) representa una relación meet.

- **Equal.** Dos regiones son iguales, si existen ambas intersecciones de límites e interiores.

$$R_{equalst}(A.B) \Leftrightarrow I(A.B) = \begin{pmatrix} \neg 0 & 0 \\ 0 & \neg 0 \end{pmatrix} \quad (4.3.4)$$

Un ejemplo de la relación equal se muestra en la Figura 4.3.2 (c).

- **Overlap.** Una región cubre parte de otra si tienen límites e interiores en común y ambos límites intersectan con el interior del otro objeto.

$$R_{overlap}(A.B) \Leftrightarrow I(A.B) = \begin{pmatrix} -0 & -0 \\ -0 & -0 \end{pmatrix} \quad (4.3.5)$$

La Figura 4.3.2 (d) presenta un ejemplo de la intersección overlap entre A y B.

- **Inside.** Una región A esta dentro de otra región B si se cumplen las siguientes condiciones:

1. A y B comparten el mismo interior pero no los límites,
2. si el límite de A es un subconjunto del interior de B
3. y el límite de B no intersecta con ninguna parte del interior de A.

$$R_{inside}(A.B) \Leftrightarrow I(A.B) = \begin{pmatrix} 0 & -0 \\ 0 & -0 \end{pmatrix} \quad (4.3.6)$$

La relación inside es presentada en la Figura 4.3.2 (e).

- **Contain.** Tres condiciones se deben cumplir para definir que una región A contiene a otra región B:

1. A y B comparten el mismo interior, pero no tienen límites en común,
2. si el límite de B es un subconjunto del interior de A,
3. y el límite de A no intersecta con el interior de B

$$R_{contain}(A.B) \Leftrightarrow I(A.B) = \begin{pmatrix} 0 & 0 \\ -0 & -0 \end{pmatrix} \quad (4.3.7)$$

La Figura 4.3.2 (f) muestra la relación cointain entre A y B.

- **Covers.** Una región A cubre a otra región B, si cumple con las siguientes condiciones:

1. Si ambas regiones comparten el mismo límite e interior,
2. si el interior de B intersecta con el límite de A
3. y si el interior de A es parte del límite de B

$$R_{covers}(A.B) \Leftrightarrow I(A.B) = \begin{pmatrix} -0 & 0 \\ -0 & -0 \end{pmatrix} \quad (4.3.8)$$

La intersección covers se muestra en la Figura 4.3.2 (g).

- **Covered by.** Una región A es cubierta por otra región B si ambas regiones tienen partes en común en sus límites e interiores. Si parte del interior de A intersectan con parte de los límites de B y si el interior de B intersecta con parte del límite de A.

$$R_{coveredby}(A.B) \Leftrightarrow I(A.B) = \begin{pmatrix} -0 & -0 \\ 0 & -0 \end{pmatrix} \quad (4.3.9)$$

La Figura 4.3.2 (h) representa la intersección covered by.

El diseño de modelos 3D topológicos está estrechamente relacionado con los requisitos específicos de la aplicación debido a su complejidad y la variación (Zlatanova et al., 2004). El diseño de reglas de construcción depende de aspectos como el tipo de objetos y tipos de relaciones permitidas; por ejemplo, sólo permitiendo triángulos o si hay objetos con inclusiones.

Dentro del algoritmo propuesto, la manera de solventar estas condicionales es por medio de la implementación de reglas semánticas denominadas *Spatial Representation Rules*.

El resultado hasta este punto es la ontología propuesta poblada con los objetos identificados asociados a su elemento del dominio más probable, además de información vinculada acerca de sus características geométricas y sus relaciones topológicas con otros elementos. Toda esta información será validada en los siguientes subprocesos.

#### 4.3.4 Validación Semántica

La validación semántica tiene como objetivo verificar semánticamente la clasificación creada. Específicamente evalúa y resuelve problemas de sobre-segmentación de elementos. La sobre-segmentación de los elementos se refiere a los casos donde un elemento real es segmentado en subcomponentes. El presente algoritmo considera dos casos de sobre-segmentación: por solapamiento y por unión.

Para mitigar la sobre-segmentación por solapamiento, el algoritmo contempla la creación e implementación de reglas semánticas llamadas *Redundant Shape Rules*.

Por otra parte, la sobre-segmentación por unión es solventada a través de su localización y unión en un nuevo segmento. Esta condición se observó solo en los elementos cilíndricos, debido a esto las reglas semánticas definidas aplican solo a dichos elementos, dichas reglas son denominadas *UnionShapenRules*.

El resultado es la ontología poblada, con correcciones semánticas aplicadas, sin embargo es necesario un proceso de validación por expertos con el fin de mejorar la calidad de la información.

Tal como se ha mencionado a lo largo de este capítulo, la implementación de todos los procesos del algoritmo dependerá del contexto en el que se aplique, esto incluye la adaptación de las reglas semánticas o incluso agregar algunas más que contribuyan a representar el comportamiento de los elementos reales.



#### 4.3.5 Validación por expertos

Esta etapa del algoritmo tiene como objetivo validar los resultados obtenidos hasta el momento. En esta etapa los expertos validan cada las clasificaciones de los elementos, sus relaciones, así como sus propiedades, si encuentran algún error, lo corrigen modificando el prototipo actual. Debido a los posibles cambios que pudieran haber surgido en las dos etapas de validación (Validación semántica y por expertos), el algoritmo vuelve a calcular todo el proceso de enriquecimiento semántico, por lo que, los resultados tanto de esta etapa, como la validación semántica se utilizan como retroalimentación. El proceso termina cuando el experto da el visto bueno al modelo semántico generado. Entonces el resultado final es una ontología poblada con información lógica, física y semántica de los elementos reconocidos y segmentados en la nube de puntos, esto es, una escena virtual enriquecida.

### 4.4 Conclusiones del Algoritmo inteligente

Esta sección describe la tercera etapa del estudio experimental, en el cual presenta el algoritmo basado en descriptores visuales y tecnología semántica, cuyo objetivo es extender el reconocimiento de primitivas combinando el uso de descriptores de características geométricas y reglas semánticas. Dicho algoritmo tendrá a capacidad de enriquecer una maqueta digital por medio de un modelo ontológico, que describe los objetos presentes en la nube de puntos incluyendo información **lógica**, **física** y **semántica**, dentro de un entorno industrial. La definición de este algoritmo inteligente contribuirá a la verificación de las hipótesis H1 y H3.



---

### Implementación: Caso de uso contexto industrial

---

De acuerdo al capítulo 3, el estudio experimental presentado para la evaluación de esta tesis ayudará a comprobar las hipótesis H1, H2 y H3 propuestas. El estudio experimental comprende tres etapas: 1)datos fuente, 2)algoritmo inteligente, 3)verificación, las cuales se desarrollarán a lo largo del presente capítulo.

En este capítulo se propone un caso de uso real con el objetivo de validar el algoritmo propuesto en la presente tesis. Como prueba de concepto, el algoritmo deberá analizar una nube de puntos de una escena real industrial y ofrecerá como resultado final una maqueta digital enriquecida, que deberá incluir un modelo ontológico con información lógica, física y semántica de los objetos pertenecientes a una escena de las instalaciones de una fábrica. Se analizará una nube de puntos, que muestra los siguientes elementos: tuberías, codos, tee y válvulas.

El algoritmo presentado en esta tesis permite la creación de maquetas digitales enriquecidas con **información lógica** mediante el cálculo de características geométricas para cada objeto tales como altura, anchura, perímetro, radio, por mencionar algunas; mientras que la **información semántica** es obtenida haciendo coincidir cada objeto con su elemento correspondiente en la ontología propuesta; finalmente la **información física** se obtiene mediante el cálculo de las relaciones espaciales de los elementos pertenecientes a la escena.

#### 5.1 Datos fuente

Esta sección corresponde a la primera etapa del estudio experimental, el análisis de datos fuente se enfoca en el estudio de las fuentes de información, esta etapa es el punto de

partida para llevar acabo la verificación de cada una de las hipótesis planteadas en esta tesis.

### 5.1.1 Nubes de puntos 3D

La captura 3D es un método mediante el cual una superficie se muestrea usando tecnología láser. El objeto o escena a capturar es analizado para obtener información sobre su forma y apariencia. La ventaja de la captura 3D por medio del escaneado láser es que puede ser posible obtener una gran cantidad de puntos con alta precisión en un periodo de tiempo relativamente corto. Debido a que los escáneres láser son instrumentos de línea de vista, es necesario realizar múltiples capturas desde diferentes estaciones para garantizar una cobertura completa de un objeto o escena (Marshall and Stutz, 2011). Los escáneres se pueden clasificar de la siguiente forma:

1. **Los escáneres de láser** se pueden diferenciar en dos grandes grupos<sup>1</sup> dependiendo si van montados o no sobre plataformas móviles: unidades dinámicas como LIDAR<sup>2</sup>, Mobile Mapping<sup>3</sup>, entre otras; y unidades estáticas. Esta última, a su vez se divide de acuerdo al sistema de medición de distancias en escáneres basados en la medida del tiempo y escáneres basados en la medida de una triangulación.
2. **Los escáneres de triangulación**<sup>4</sup>, son de corto alcance es decir, su alcance es inferior a 10 metros, y alta precisión, del orden de micras. Son utilizados comúnmente para documentar objetos de dimensiones reducidas, las cuales exigen un muestreo con un alto grado de detalle y precisión (Fechteler et al., 2007).
3. **Los escáneres de medida de tiempo**, con este tipo de escáner es posible calcular la distancia o profundidad del objeto o escena midiendo el tiempo que transcurre entre la emisión del pulso láser y la recepción, después de que éste sea reflejado por el objeto. Son utilizados principalmente en la documentación de edificios, tanto en trabajos exteriores como interiores (Nuttens et al., 2012).

El resultado de estos muestreos es una nube de puntos. Una nube de puntos es un conjunto de puntos no organizados que representan en una forma discreta el objeto o escena en cuestión y se caracterizan por sus coordenadas espaciales (Erdos et al., 2014), dicha nube de puntos puede o no contener uno o más canales de datos en cada punto como color, profundidad, iluminación, oclusión, zona, por mencionar algunos. Además de tener características tales como datos no uniformes (proveniente de una integración de diferentes conjuntos de datos), densidad inconstante, errores de alineación, ruidos,

---

<sup>1</sup>[http://www.innovmetric.com/polyworks/3D-scanners/pdf/3D\\_Scanning\\_Options-Desktop\\_Engineering-January\\_2011.pdf](http://www.innovmetric.com/polyworks/3D-scanners/pdf/3D_Scanning_Options-Desktop_Engineering-January_2011.pdf)

<sup>2</sup><http://www.lidar.co.uk/>

<sup>3</sup><http://www.3dlasermapping.com/products/mobile-scanning>

<sup>4</sup><http://2tthtalk.com/images/scanst.pdf>

datos desordenados, etc (Sitek et al., 2006). La nube de puntos, dependiendo la escena muestreada, puede contener varios cientos de millones de puntos. El conjunto de datos que se utilizó para el desarrollo de la validación y las pruebas de esta tesis contiene un poco más de setenta millones de puntos representados solo por coordenadas atómicas. La Figura 3.3.2 muestra una vista de 360 grados de la nube de puntos, la cual se refiere a las instalaciones reales de una fábrica, donde se muestran elementos tales como tuberías, codos, té y válvulas.

La nube de puntos está almacenada en un fichero con un formato XYZ. Estos archivos contienen las coordenadas atómicas, es decir, un conjunto de líneas con coordenadas  $x$ ,  $y$  y  $z$ , separados por espacios, tabulaciones o comas. La lista 5.1 muestra un extracto del fichero original.

Lista 5.1: Extracto de fichero xyz de nube de puntos

1	89.302322	10.968063	0.306595
2	89.301071	10.939529	0.296555
3	89.311569	10.940933	0.296432
4	89.306870	10.942551	0.295456
5	89.300980	10.943527	0.294296
6	89.288589	10.941177	0.295975
7	89.294113	10.939987	0.294815
8	89.292435	10.943161	0.293533
9	89.283218	10.942490	0.291489
10	89.291672	10.946823	0.292068
11	89.288162	10.949051	0.293076
12	89.285751	10.947830	0.296280
13	89.286240	10.952164	0.294662
14	89.283127	10.954605	0.296585
15	89.289017	10.957626	0.295425
16	89.284378	10.963364	0.294571
17	89.298630	10.938278	0.299759
18	89.309219	10.951797	0.299057
19	89.283035	10.942368	0.305344
20	89.292740	10.939255	0.298599
21	89.284744	10.943253	0.301743
22	89.286545	10.940109	0.299362
23	89.285416	10.943588	0.297470
24	89.282669	10.946213	0.303665
25	89.285385	10.947647	0.300735
26	89.283829	10.950943	0.297806
27	89.294998	10.938309	0.310715
28	89.281967	10.947617	0.310013
29	89.282303	10.946030	0.308121
30	89.287918	10.953018	0.304977
31	89.282089	10.950027	0.306107
32	89.281998	10.945389	0.309036
33	89.281418	10.949661	0.310318
34	89.299271	10.958847	0.297897

35	89.304276	10.957352	0.298203
36	89.035873	10.874741	0.289566
37	89.037552	10.871353	0.290085
38	89.047226	10.867752	0.297195
39	89.044113	10.866165	0.300217
40	89.032700	10.864609	0.297623

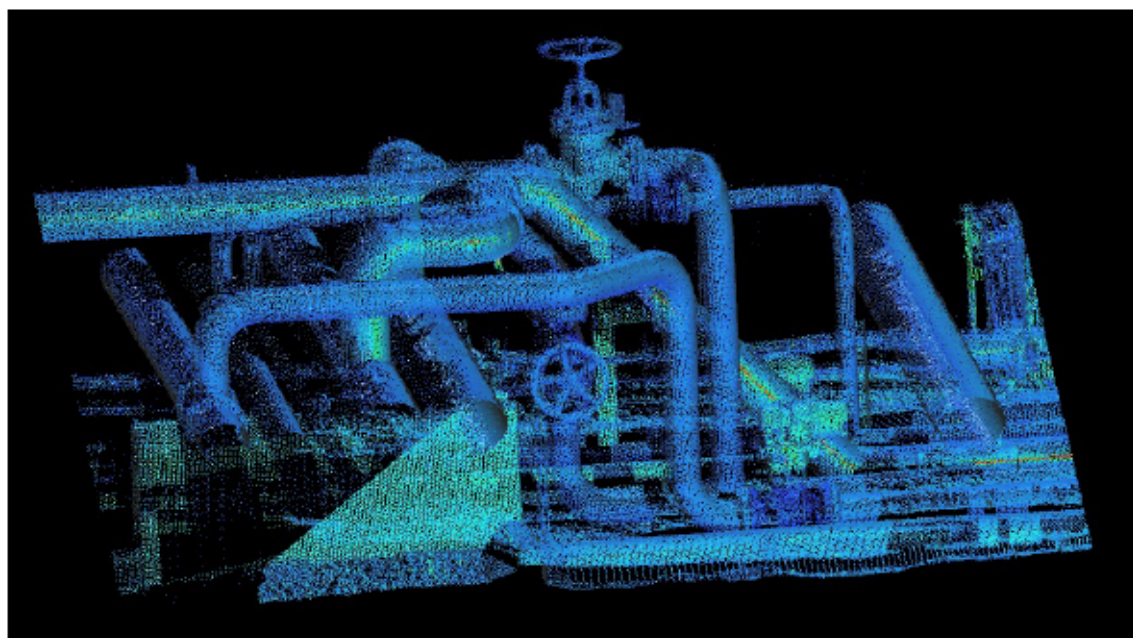


Fig. 5.1.1: nube de puntos desde el interior de una fábrica

La Figura 5.1.1 representa una vista de 360° de un segmento de la nube de puntos de una instalación industrial real, la cual tiene un total de 437.625 puntos. Dentro de esta nube de puntos existen elementos que se asemejan a las primitivas geométricas tales como cilindros, esferas y toros que deberán ser reconocidos y debidamente etiquetados por el algoritmo, como segundo paso, la información lógica, física y semántica de los objetos reconocidos deberá ser calculada y almacenada en el modelo ontológico propuesto. Con esto en mente, el algoritmo se puede dividir en dos procesos: el reconocimiento primitivas y enriquecimiento semántico, tal y como se ha presentado en el Figura 4.1.1.

## 5.2 Implementación del algoritmo propuesto

### 5.2.1 Proceso de reconocimiento de primitivas

El proceso de reconocimiento de primitivas pretende responder a la pregunta *¿Existe algún objeto conocido dentro de la escena escaneada?*. Para lo cual es necesario proveer como *entrada* la **nube de puntos** descrita. La *salida* deberá ser una **lista de objetos clasificados** de acuerdo a sus características geométricas.

Como ya se ha mencionado, este proceso se divide en 3 tareas:

1. Pre-proceso
2. Extracción de descriptores
3. Clasificación

#### 5.2.1.1 Pre-proceso

El pre-proceso busca mejorar la calidad de la información antes de ser analizada por el extractor de descriptores, eliminando posibles ruidos o información redundante conocida. La implementación del pre-proceso da como resultado una nube de puntos sin ruido, sin coordenadas redundantes, y ordenada mediante una estructura jerárquica. Además se calculó la normal de los puntos, para su uso posterior en los siguientes módulos.

#### 5.2.1.2 Extracción de descriptores

La extracción de descriptores pretende representar primitivas geométricas a través de características específicas descritas matemáticamente.

Las primitivas geométricas seleccionadas para este caso de uso son cilindro, toro y esfera. Debido a que los cilindros son una de las primitivas geométricas que se encuentran más frecuentemente en escenas industriales. Existen estudios como Hakala et al. (1981); Rabbani and Heuvel (2005) que mencionan que el 85% de los objetos encontrados en escenas industriales corresponden a elementos que se asemejan a planos, esferas y cilindros (Hakala et al., 1981; Petitjean, 2002; Rabbani and Heuvel, 2005). Sin embargo, si se incluyen las primitivas toroidales, el porcentaje se eleva al 95% (Rabbani and Heuvel, 2005; Requicha and Voelcker, 1982), esto demuestra la importancia del reconocimiento de primitivas geométricas en escenas industriales. Particularmente el reconocimiento de cilindros, los cuales respaldan la importancia de su estudio en la gran cantidad de investigaciones enfocadas a ellos, entre las más sobresalientes están Attene, Falcidieno, et al. (2006); Attene and Patane (2010); Rabbani and Heuvel (2005); Schnabel et al. (2007); Tran et al. (2015); Vosselman and Gorte (2004).

A continuación se describe cómo calcular los parámetros de la primitiva geométrica cilindro, el cual utilizamos para implementar y probar el proceso de reconocimiento de primitivas.

Un cilindro se identifica convenientemente a través del algoritmo de mejor ajuste, tal y como se muestra en el Algoritmo 1 y la Figura 5.2.1, el cual tiene como *entrada* una nube con un número finito de puntos, y los vectores normales correspondientes a cada punto. Los enfoques de la literatura revisada Attene, Falcidieno, et al. (2006); Attene and Patane (2010); Chaperon et al. (2001); Kobbelt (2005); Pauly et al. (2002); Rabbani and Heuvel (2005); Varady et al. (1997) inspiraron la descripción de este algoritmo.

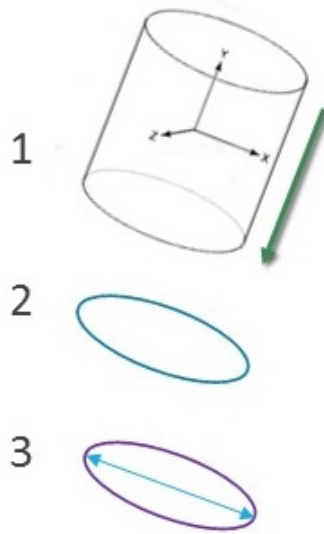


Fig. 5.2.1: Algoritmo mejor ajuste de Cilindro

---

**Script 1** Algoritmo mejor ajuste de Cilindro

---

- 1: Encontrar la orientación del eje del cilindro  $\vec{a}$  calculando el vector propio correspondiente al valor propio más pequeño de la matriz de covarianza  $C$  de los vectores normales
  - 2: Proyectar los puntos sobre el plano con normal  $\vec{a}$  y pasando a través del origen
  - 3: Ajustar la proyección 2D a un círculo para encontrar el radio  $r$  y un punto  $c$  perteneciente al eje (centro del cilindro)
- 

Los parámetros que definen un cilindro son: 1) un conjunto finito de puntos que pertenecen a la superficie  $P = \{p_i\}_{i=1}^N$ , 2) el radio  $r$ , 3) un punto sobre el eje  $c$  y 4) un vector normal que defina el eje del cilindro  $\vec{a}$ . Dos de estos parámetros serán valores de *entrada* para el algoritmo de mejor ajuste del cilindro los cuales son el conjunto finito de puntos y el vector de normales de cada punto, este último fue calculado en el pre-proceso. Mientras que el resto de los parámetros serán valores de *salida* del algoritmo.

Como primer paso se calcula la orientación del eje del cilindro obteniendo el vector



propio más ortogonal a todos los vectores normales, es decir, el vector propio  $\vec{a}$  que corresponda al valor propio más pequeño de la matriz de covarianza  $C$  de los vectores normales. Definida por 5.2.1.

$$C = \sum_{i=1}^{n_{cluster}} (n_i n_i^T) \quad (5.2.1)$$

donde  $n_i$  es una fila del vector normal. Los otros dos vectores propios, a los que llamaremos  $V_X$  y  $V_Y$ , crearán un sistema de coordenadas en el plano para hacer la proyección 2D del círculo.

Como segundo paso se debe proyectar los puntos sobre el plano  $\tilde{p}_i = [\langle p_i, V_X \rangle_2, \langle p_i, V_Y \rangle_2]$  con normal  $\vec{a}$  y pasando a través del origen  $O$ . Estos puntos proyectados describen un círculo en el plano. Por lo que el siguiente paso es calcular el centro  $c$  y el radio  $r$ . Esta operación se puede reducir al problema bi-dimensional de mejor-ajuste del círculo [13]

El cálculo de los parámetros radio  $r$  y centro  $c$  en  $c = (c_x, c_y, c_z)$  por medio del mejor ajuste de mínimos cuadrados del cilindro a un conjunto de puntos 3D se formaliza de la siguiente manera: Sea  $P$  un conjunto de  $N$  puntos  $(x_i, y_i, z_i)$ , y sea

$$x^2 + y^2 + z^2 + c_x^2 + c_y^2 + c_z^2 - 2c_x x - 2c_y y - 2c_z z - r^2 = 0 \quad (5.2.2)$$

la cual, en forma de vector, equivale a:

$$[2x \ 2y \ 2z \ 1] \begin{bmatrix} c_x \\ c_y \\ c_z \\ r^2 - c_x^2 - c_y^2 - c_z^2 \end{bmatrix} = x^2 + y^2 + z^2 \quad (5.2.3)$$

donde los cuatro valores desconocidos  $c_x, c_y, c_z$  y  $r$  han sido aislados en un único vector. Por lo que, sustituyendo  $x, y$  y  $z$  con las coordenadas de los puntos en  $P$  se obtiene el siguiente sistema lineal sobredeterminado:  $Aw = b$  donde

$$A = \begin{bmatrix} 2x_1 & 2y_1 & 2z_1 & 1 \\ 2x_2 & 2y_2 & 2z_2 & 1 \\ \vdots & & & \\ 2x_n & 2y_n & 2z_n & 1 \end{bmatrix}, b = \begin{bmatrix} x_1^2 & y_1^2 & z_1^2 \\ x_2^2 & y_2^2 & z_2^2 \\ \vdots & & \\ x_n^2 & y_n^2 & z_n^2 \end{bmatrix} \text{ and } w = \begin{bmatrix} c_x \\ c_y \\ c_z \\ r^2 - c_x^2 - c_y^2 - c_z^2 \end{bmatrix} \quad (5.2.4)$$

los cuales se pueden resolver en el sentido de mínimos cuadrados calculando  $w = (A^T A)^{-1} A^T b$ .

Después de haber establecido los parámetros  $n, c$  y  $r$  del cilindro, se calcula el error ajuste L2 por medio de:

$$L^2 = \sum_{i=1}^n a(V_i) (\|(V_i - c) \times n\|_2 - r)^2 \quad (5.2.5)$$

El resultado de la implementación de la extracción de descriptores en el algoritmo propuesto es una lista de índices de semejanza con cada una de las primitivas geométricas predefinidas, con los cuales se calcula el error de ajuste.

### 5.2.1.3 Clasificación

Para la implementación del proceso de clasificación se analizaron tres algoritmos clásicos de reconocimiento de primitivas con el fin de seleccionar aquel que se adapte mejor a las necesidades del algoritmo. Tal como se ha mencionado, las necesidades a cumplir son:

1. Tener la capacidad de procesar y segmentar nubes de puntos.
2. Tener la capacidad de reconocer primitivas geométricas, específicamente cilindros, toros y esferas.
3. Calidad en el segmentado y clasificación de las primitivas geométricas.

Plumber(Mortara and Patané, 2004), HFP(Attene and Patane, 2010) y RansacSD(Schnabel et al., 2007) son los algoritmos seleccionados para su análisis debido a que cumplen con los dos primeros requisitos, siendo el tercer requisito el motivo de una evaluación de funcionalidad. La importancia de estos algoritmos puede hacerse notar en los múltiples trabajos en los que son mencionados, por ejemplo, Plumber ha sido objeto de diversos estudios importantes, como Attene, Katz, et al. (2006); Attene and Robbiano (2009); Mortara, Patané, and Spagnuolo (2006); Shamir (2006), HFP se ha mencionado en diversas publicaciones, incluyendo: Attene, Katz, et al. (2006); X. Chen et al. (2009); Golovinskiy and Funkhouser (2008), y finalmente la relevancia de RansacSD se mencionó anteriormente por Haala and Kada (2010); Y. Li, X. Wu, et al. (2011); Nan et al. (2010); R. B. Rusu et al. (2009).

Estos algoritmos se describen brevemente a continuación:

- *Plumber: Mesh segmentation into tubular parts*(Mortara and Patané, 2004). Este algoritmo está basado en el análisis de formas multi-escala, Plumber descompone las formas en características tubulares y al mismo tiempo extrae el eje del esqueleto de dichas características. Las partes alargadas de la forma son capturados por las características tubulares.
- *HFP: Hierarchical mesh segmentation based on fitting primitives*. HFP es un algoritmo basado en una agrupación jerárquica de los elementos, el cual fue desarrollado por primera vez para mallas de triángulos por Attene and Patane (2010) y luego ampliada y adaptada para nubes de puntos no estructurados por Attene and Patane (2010). HFP segmenta jerárquicamente la nube de puntos en grupos que mejor se ajuste a un conjunto predefinido de primitivas. Inicialmente, cada punto representa un solo grupo; después, para cada iteración se consideran todos los pares de grupos

adyacentes y la pareja cuyos puntos pueden ser mejor ajuste por una de las primitivas son fusionados en un solo clúster nuevo.

- *RansacSD* fue seleccionado para el análisis debido a su relevancia en el ámbito del reconocimiento de formas primitivas dentro de una nube de puntos.

HPF y RansacSD tienen la capacidad de reconocer las diferentes elementos que se asemejan a las primitivas geométricas tales como cilindros, planos, toros, esferas y conos, sin embargo Plumber reconoce solo elementos con una forma tubular. Los tres algoritmos fueron sometidos a una evaluación con dos casos de uso, los cuales tenían como objetivo segmentar y clasificar elementos pertenecientes a una nube de puntos a partir de las características geométricas de los mismos.

Las características de las nubes de puntos analizadas para la evaluación se describen a continuación:

La nube de puntos original representa una vista de 360<sup>a</sup> del interior de una planta industrial real, a partir de ésta se extrajeron tres segmentos con 10352, 44996 y 143663 puntos. Estas secciones están conformadas por elementos (es decir, subconjuntos de los puntos) que se asemejan a primitivas geométricas tales como formas tubulares, planos y toros, los cuales serán reconocidos y debidamente etiquetados por HPF, Plumber y RansacSD según sus características. La Tabla 5.2.1 muestra una descripción de los elementos que componen cada nube de puntos a analizar.

Tabla 5.2.1: Elementos existentes en las nubes de puntos evaluadas

<i>Nube de puntos</i>	<i>Caso 1</i>	<i>Caso 2</i>		
	Elem Tub	Planos	Toros	Esferas
G	18	3	2	1
M	6	0	5	1
P	6	7	0	0

Los dos caso de uso a evaluar difieren de acuerdo a dos variables: tipos de primitivas a reconocer y características de la nube de puntos.

1. Primitivas para reconocer. El primer caso de uso analiza solo elementos tubulares (Elem Tub), mientras que el segundo analiza planos, esferas y toros.
2. Características de la nube de puntos. Se presentan tres nubes de puntos con diferentes características: el tamaño (número de puntos), el tipo de densidad y si tiene algún tipo de pre-procesamiento aplicado a la nube de puntos. La Tabla 5.2.2 muestra una descripción de cada variable.

Tabla 5.2.2: Descripción de las características de las nubes de puntos

<i>Característica</i>	<i>Valor</i>	<i>Descripción</i>
Tamaño	143663 puntos (G) 44996 puntos (M) 10352 puntos (P)	El tamaño de la nube de puntos se considera grande (G), mediana (M) o pequeña (P) de acuerdo con el número de puntos que lo componen. Esta nube de puntos representa un segmento del interior de una fábrica real, que contiene tubos, codos, y planos.
Densidad	Constante (C) Inconstante (I)	Una nube de puntos tiene una densidad constante (C) cuando se tiene el mismo número de puntos por unidad de medida a lo largo de la misma. Mientras que, se dice que una nube de puntos tiene una densidad inconstante (I) cuando la densidad por unidad de medida varía.
Pre-procesamiento	Si (S) No (N)	Los métodos pre-procesamiento son los métodos aplicables a la nube de puntos para el correcto funcionamiento del algoritmo y pueden ser diversos.

La información obtenida a partir de los resultados de cada algoritmo fueron medidos a través de los puntos analizados, puntos perdidos, tiempo de procesamiento, segmentos generados, segmentos etiquetados correctamente y errores de marcado.

El primer caso de uso tenía el objetivo de reconocer los elementos tubulares con cada algoritmo en tres nubes de puntos: grande (G), mediana (M) y pequeña (P), con una densidad inconstante y dos valores para el parámetro de pre-procesamiento. El rendimiento de los algoritmos HFP, Plumber y RansacSD para el reconocimiento de elementos tubulares se muestran en la Tabla 5.2.3

Tabla 5.2.3: Rendimiento de los algoritmos de HFP, Plumber y RansacSD en el reconocimiento de elementos tubulares

<i>Algoritmo</i>	<i>Parámetros de entrada</i>			<i>Resultado de las pruebas</i>			
	<i>Nube de puntos</i>			<i>Primitivas</i>	<i>% Puntos analizados</i>	<i>% Puntos perdidos</i>	<i>Promedio de tiempo (Seg.)</i>
	<i>Tamaño</i>	<i>Densidad</i>	<i>Pre-procesamiento</i>				
<i>HFP</i>	G	I	No	Elem Tub	71%	29%	63
<i>RansacSD</i>	G	I	No	Elem Tub	91%	9%	12
<i>Plumber</i>	G	C	Si	Elem Tub	N/A	N/A	N/A
<i>HFP</i>	M	I	No	Elem Tub	88%	12%	54
<i>RansacSD</i>	M	I	No	Elem Tub	67%	33%	10
<i>Plumber</i>	M	C	Si	Elem Tub	N/A	N/A	N/A

Continúa en la siguiente página

Tabla 5.2.3 – continuación de página previa

Algoritmo	Parámetros de entrada				Resultado de las pruebas		
	Nube de puntos			Primitivas	% Puntos analizados	% Puntos perdidos	Promedio de tiempo (Seg.)
	Tamaño	Densidad	Pre-procesamiento				
HFP	P	I	No	Elem Tub	88%	12%	1
RansacSD	P	I	No	Elem Tub	85%	15%	0,8
Plumber	P	C	Si	Elem Tub	66%	34%	380

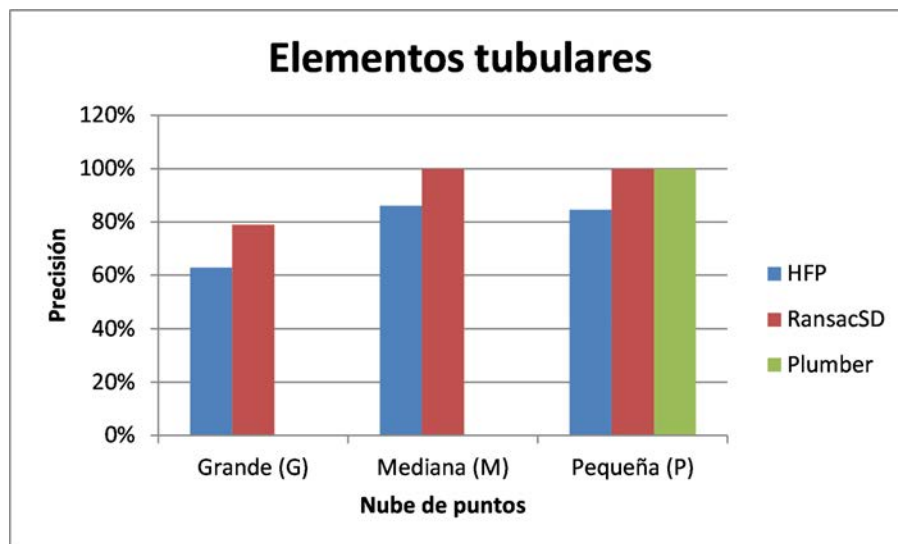


Fig. 5.2.2: Precisión de los algoritmos de HFP, Plumber y RansacSD en el reconocimiento de elementos tubulares

La Figura 5.2.2 muestran el alto rendimiento de RansacSD y Plumber en el reconocimiento de objetos que se asemejan a elementos tubulares alcanzando una precisión de hasta el 100%. En los dos primeros casos, Plumber no se evaluó porque las condiciones de los datos no cumplían con los requisitos mínimos para el algoritmo, que son: una nube de puntos con densidad constante y con un número de puntos menor a 20000. Sin embargo, en la tercera prueba, Plumber logra una precisión del 100% en el reconocimiento de elementos tubulares. Esta prueba utiliza una nube de puntos de tamaño pequeño con un método de pre-procesamiento, que contiene los filtros necesarios para la generación de una densidad constante en la de nube de puntos fuente. Mientras tanto, HFP y RansacSD mantuvieron un porcentaje de precisión entre 63% y 100% durante las pruebas, que muestra un



rendimiento mejorado para la nube de puntos pequeña. Aunque HFP no obtuvo 100% de precisión en el reconocimiento de objetos que se asemejan a elementos tubulares, la Tabla 5.2.3 muestra que la calidad de los resultados puede ser considerado como más alto porque HFP analizado hasta 88% de los puntos y tenía una pérdida máxima de información de 29%, a diferencia de Plumber y RansacSD, que tenía pérdidas de puntos de hasta un 34%. Por otra parte, RansacSD resultó ser 98% más rápido que Plumber y hasta 80% más rápido que HFP. Esto se debe a RansacSD realiza menos iteraciones que los otros algoritmos.

Mientras tanto, el segundo caso tiene el objetivo de reconocer elementos que se asemejen a planos, esferas y toros, utilizando los algoritmos RansacSD y HFP en tres nubes de puntos con tamaños diferentes: grande, mediana y pequeña, con una densidad inconstante y sin pre-procesamiento.

Tabla 5.2.4: Rendimiento de HFP y RansacSD en el reconocimiento de objetos que se asemejan planos, esferas y toros

Algoritmo	Parámetros de entrada			Resultado de las pruebas			
	Nube de puntos			Primitivas	% Puntos analizados	% Puntos perdidos	Promedio de tiempo (Seg.)
	Tamaño	Densidad	Pre-procesamiento				
HFP	G	I	No	Planos, esferas y toros	71%	29%	252
RansacSD	G	I	No	Planos, esferas y toros	91%	9%	48
HFP	M	I	No	Planos, esferas y toros	88%	12%	216
RansacSD	M	I	No		67%	33%	8
HFP	P	I	No	Planos, esferas y toros	88%	12%	4
RansacSD	P	I	No	Planos, esferas y toros	85%	15%	3,2

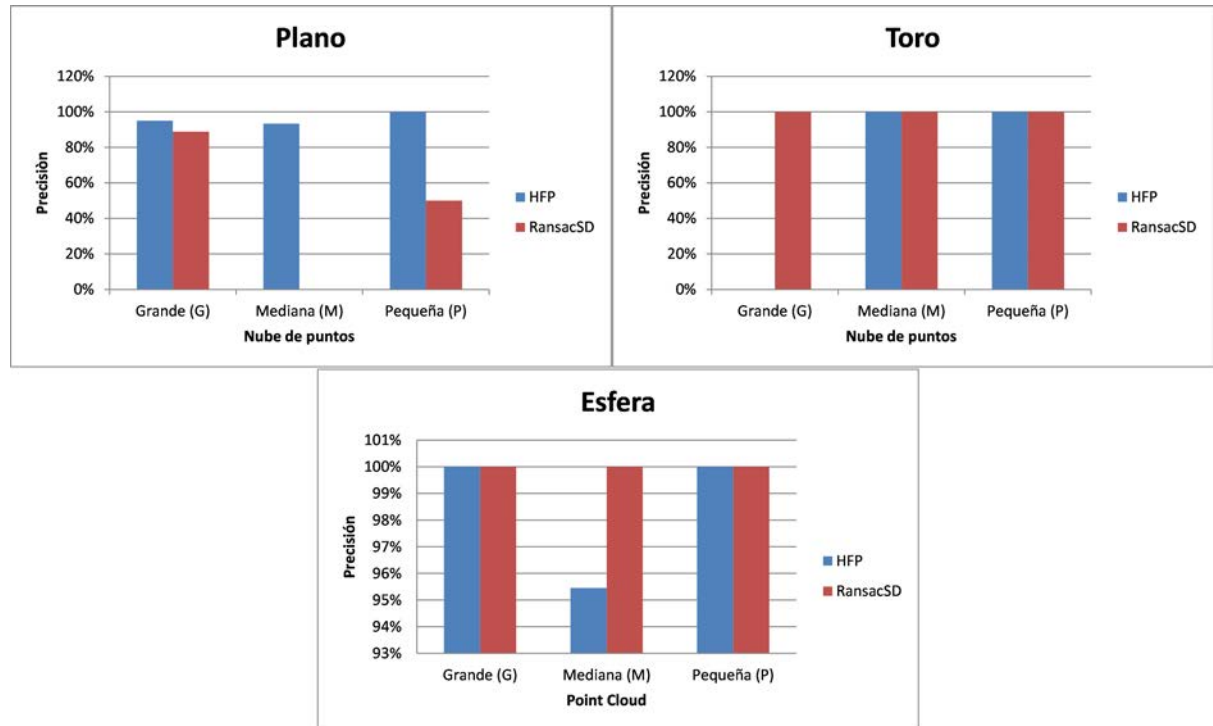


Fig. 5.2.3: Precisión de HFP y RansacSD en el reconocimiento de objetos que se asemejan planos, esferas y toros en una nube de puntos

El rendimiento de HFP y RansacSD en el reconocimiento de objetos que se asemejan planos, esferas y toros en la prueba se muestra en la Tabla 5.2.4 y la Figura 5.2.3. El tiempo de procesamiento de ambos algoritmos a través de diferentes nubes de puntos es consistente con el tamaño de la nube de puntos. Podemos ver que cuanto más grande es la nube de puntos, mayor es el tiempo de procesamiento, aunque la diferencia en el tiempo entre los dos algoritmos muestra que RansacSD es más rápido. Ambos algoritmos tienen una pérdida de puntos de entre el 9% y el 33%, por lo que se puede observar que mejora la calidad del análisis al considerar un mayor número de puntos.

En reconocimiento de los planos, es importante mencionar que la nube de puntos mediana no contiene planos, por lo que no se consideró en la prueba. Sin embargo en el caso de las nubes de puntos de tamaño grande y pequeña, HFP tuvo un promedio de 97% de precisión y mantiene su comportamiento constante a través de las nubes de puntos, mientras que RansacSD alcanzó hasta un 50% de precisión en la nube de puntos de tamaño pequeño, pero muestra una mejora notable en el caso de la nube de puntos grande. En el caso de reconocimiento de toros para nubes de puntos de tamaño grande y mediano, RansacSD obtuvo 100% de precisión para los dos casos de estudio. Por el contrario, HFP no obtuvo ningún resultado al reconocer toros en las nubes de puntos. Por último, en el

reconocimiento de los elementos que se asemejan a esferas en nubes de puntos grandes y medianas, HFP y RansacSD logran una precisión alta, HFP tiene un promedio de 97,72%, mientras que RansacSD logró un 100% precisión pero sólo en la nube de puntos de tamaño grande.

Como podemos observar HFP mostró un comportamiento constante a lo largo de los casos de uso y todas las primitivas geométricas, con esto en mente, y debido a que se realizó una estancia de investigación en el *National Research Council*, en *The Institute for Applied Mathematics and Information Technologies*<sup>5</sup>, donde fue desarrollado HFP, se decidió seleccionarlo para la implementación del algoritmo propuesto en esta tesis. Por otra parte, RansacSD se utilizará para el proceso de evaluación desarrollado en el Capítulo 6, con lo cual comprobaremos la generalidad de este algoritmo ya que tiene la capacidad de soportar diferentes librerías con características similares, de forma que permite modificar los alcances del algoritmo acuerdo a las necesidades de aplicación.

El uso de la estructura de árbol *kd* para la organización de la nube de puntos, permite realizar una segmentación jerárquica, de forma que se evalúan regiones conectadas aproximándolas a un conjunto predefinido de primitivas. Este conjunto de primitivas es arbitrario, y no influye en la validez del proceso.

Cada primitiva se identifica por sus parámetros, tal como lo se detalló en la sección anterior con el cilindro. El cálculo de los parámetros para cada primitiva se utiliza para identificar la primitiva y también para obtener la distancia entre el conjunto de puntos en el clúster y la primitiva.

Inicialmente, cada punto representa un solo grupo; después, para cada iteración se consideran todos los pares de grupos adyacentes y la pareja cuyos puntos decanten su mejor ajuste por la misma primitiva, se fusionan en un solo y nuevo clúster.

En general la fórmula del error de ajuste es expresada como la suma  $L^2_{primitiva}(P)$  de los residuos cuadrados entre  $P$  y su mejor ajuste a la primitiva evaluada.

Por lo que, para cada tipo de primitiva se calculan los parámetros de ajuste correspondientes y se evalúa el error de aproximación, es decir, si se requiere un algoritmo para el reconocimiento de planos y cilindros, entonces es necesario calcular 1) los parámetros para el mejor ajuste de un plano y 2) los parámetros para el mejor ajuste de cilindros, de manera que posteriormente se evalúen los error de aproximación tanto del plano como del cilindro con respecto al conjunto de puntos evaluados. Entonces el valor mínimo es considerado para volver a ordenar la estructura.

Como resultado final se genera una clasificación simple de los elementos, es decir, una lista de grupos de puntos (segmentos) clasificados por primitivas de acuerdo a su error de ajuste mínimo. Dicha lista permite responder a la pregunta inicial que resolvería este proceso de reconocimiento de primitivas: *¿Existe algún objeto conocido dentro de la escena escaneada?*

Sin embargo esta lista clasificada no es suficiente para los objetivos de esta tesis de-

---

<sup>5</sup><http://www.imati.cnr.it/>



bido a que pueden presentarse errores de clasificación como solapamiento de segmentos, segmentación múltiple correspondiente a un mismo objeto, por nombrar algunos; y el sentido semántico alcanzado hasta este momento consta solo de un etiquetado simple sin un sentido semántico formal.

#### 5.2.1.4 Resultados del proceso de reconocimiento de primitivas

En este proceso, la escena fue segmentada por el algoritmo de reconocimiento primitivas, clasificando los objetos en función de sus características geométricas, esto es, se ha creado la representación lógica de la escena. La Tabla 5.2.5 muestra los segmentos clasificados de la nube de puntos, donde cada segmento tiene asignado una clase de primitiva. Esta tabla muestra 57 segmentos etiquetados de acuerdo a sus características geométricas, donde *:Shape* describe la forma como punto de referencia; *GeometricPrimitive:Shape* describe el tipo de geométrica primitiva clasificada; y *:Shape.Points* define el número de puntos que conforman la forma.

Tabla 5.2.5: Resultados del proceso de reconocimiento de primitivas

<i>:Shape</i>	<i>Geometric Primitive</i>	<i>:Shape.Points</i>
shape_001	Cylinder	57,472
shape_002	Cylinder	30,619
shape_003	Cylinder	18,705
shape_004	Cylinder	15,158
shape_005	Cylinder	10,623
shape_006	Cylinder	10,019
shape_007	Cylinder	10,715
shape_008	Cylinder	10,135
shape_009	Cylinder	8,519
shape_010	Cylinder	5,812
shape_011	Sphere	6,560
shape_012	Cylinder	5,014
shape_013	Torus	5,454
shape_014	Torus	4,709
shape_015	Cylinder	4,313
shape_016	Cylinder	3,979
shape_017	Torus	4,609
shape_018	Sphere	3,580
shape_019	Torus	3,292
shape_020	Torus	3,140
Continúa en la siguiente página		

Tabla 5.2.5 – continuación de página previa

<i>:Shape</i>	<i>Geometric Primitive</i>	<i>:Shape.Points</i>
shape_021	Torus	3,299
shape_022	Cylinder	3,028
shape_023	Cylinder	2,709
shape_024	Cylinder	3,757
shape_025	Cylinder	3,227
shape_026	Sphere	2,840
shape_027	Cylinder	2,066
shape_028	Sphere	2,879
shape_029	Cylinder	2,002
shape_030	Cylinder	2,981
shape_032	Cylinder	1,737
shape_033	Cylinder	1,620
shape_034	Torus	1,598
shape_035	Torus	1,696
shape_036	Cylinder	1,777
shape_037	Torus	2,448
shape_038	Cylinder	1,846
shape_039	Cylinder	1,834
shape_040	Cylinder	1,400
shape_041	Cylinder	1,127
shape_042	Cylinder	1,863
shape_043	Torus	1,027
shape_044	Cylinder	956
shape_045	Cylinder	156
shape_046	Cylinder	1,291
shape_047	Sphere	824
shape_048	Torus	856
shape_049	Torus	1,042
shape_050	Torus	1,222
shape_051	Cylinder	753
shape_052	Sphere	137
shape_053	Cylinder	779
shape_054	Torus	674
shape_055	Torus	718
shape_056	Cylinder	642
shape_057	Cylinder	923
shape_058	Torus	594

La Tabla 5.2.6 muestra un resumen de los segmentos obtenidos por el algoritmo como respuesta a la pregunta inicial *¿Existe algún objeto conocido dentro de la escena escaneada?*, por lo que se puede concluir que para la nube de puntos seleccionada si existen objetos conocidos, 36 de ellos son cilindros, 10 toros y 11 esferas, las cuales serán analizadas para conocer su sentido semántico en el siguiente proceso del algoritmo.

Tabla 5.2.6: Segmentos clasificados

<i>Nube de puntos original</i>	<i>Concepto</i>	<i>Resultados</i>
<i>Tamaño (# de puntos)</i>	<i>Primitiva</i>	<i>Total de segmentos</i>
437625	Cylinder	36
	Torus	10
	Sphere	11

### 5.2.2 Proceso de enriquecimiento semántico

Este proceso tiene como objetivo calcular la información semántica y física de cada objeto reconocido. Pretende resolver la pregunta *¿Qué tipo de elementos tiene la escena escaneada?*, para lo cual es necesario proporcionar como *entrada* una lista de segmentos clasificados en el proceso anterior de acuerdo a sus características geométricas. La *salida* esperada será un conjunto de datos lógicos, físicos y semánticos que describen los objetos que componen la escena real.

Cada elemento clasificado será vinculado a una ontología con el fin de proporcionar un significado semántico, de esta forma se enriquecerá maqueta la maqueta digital.

#### 5.2.2.1 Ontología propuesta

El diseño de una ontología no es una tarea simple, a diferencia de otros mecanismos de descripción, una ontología requiere de un profundo conocimiento del área de aplicación en la que se pretende introducir la ontología. Dicho conocimiento conducirá a definir de una manera correcta el significado de sus elementos, así como sus relaciones. La ontología deberá representar fielmente la realidad, tener la capacidad de ser reutilizado y de poder extrapolarse a ambientes semejantes.

En este sentido una ontología de dominio fue desarrollada con el fin de unir elementos de lenguaje natural al algoritmo de enriquecimiento de escenas.

La metodología de Methontology Gómez-Pérez (1998) fue la seleccionada para crear la ontología. Como se ha descrito en el estado del arte esta metodología propone un ciclo de vida de construcción de ontologías basado en prototipos *en evolución*, ya que permite

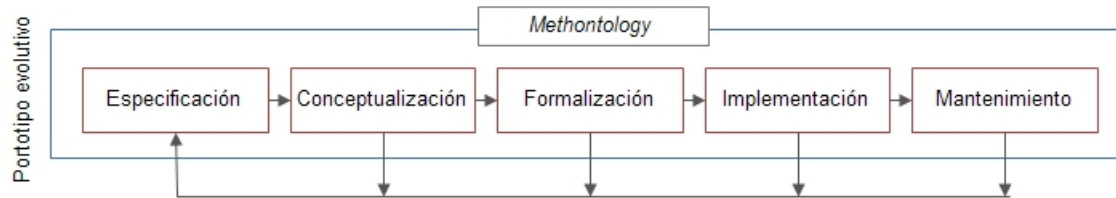


Fig. 5.2.4: Ciclo de vida de la construcción de una ontología de acuerdo con la metodología Methontology (Gómez-Pérez, 1998)

agregar, cambiar y eliminar términos en cada nueva versión y se adapta perfectamente a los objetivos de esta tesis los cuales pretenden que el algoritmo evolucione hacia diferentes áreas, la Figura 5.2.4 muestra el ciclo de vida de la construcción de una Ontología de acuerdo con la metodología Methontology. Los pasos a seguir para la construcción de una ontología estecificados por Methontology son los siguientes:

1. **Especificación.** Tiene el objetivo de explicar los motivos por los cuales se está construyendo la ontología, cuáles son sus usos previstos, quiénes son los usuarios finales y cuales son los requisitos deben cumplir con la ontología (Suárez-Figueroa, 2009). Técnicamente este proceso consiste en responder a las siguientes preguntas: ¿Cuál es el dominio de la ontología?, ¿Para qué se usara la ontología?, ¿Para qué tipos de preguntas la información en la ontología debería proveer respuestas?, ¿Quién usara y mantendrá la ontología? (T. Berners-Lee et al., 2001). Por lo que, la ontología a desarrollar deberá permitir el intercambio de información a través de especificaciones orientadas a la definición de términos, vocabularios y glosarios, todas ellas mapeadas en algún lenguaje ejecutable que facilite su implementación. Considerando lo anterior, las respuestas a los cuestionamientos fueron los siguientes:
  - *¿Cuál es el dominio de la ontología?*. El dominio de la ontología va dirigido a conceptos relacionados con elementos que se pueden encontrar en instalaciones industriales.
  - *¿Para qué se usará la ontología?*. Para manipular información, conceptos y vocabulario de elementos qe describan escenas virtuales de instalaciones industriales.
  - *¿Para qué tipos de preguntas la información en la ontología debería proveer respuestas?*. La ontología debería tener la capacidad de contestar preguntas que describan lógica, física y semánticamente los elementos pertenecientes a una escena virtual de instalaciones industriales. Entre las preguntas más relevantes están:
    - ¿Qué objeto está en la posición x, y, z?

- ¿Qué características físicas tiene el elemento de la posición  $x, y, z$ ?
- ¿Quién es el posible fabricante del objeto en la posición  $x, y, z$ ?
- ¿Qué tan alto es el objeto de la posición  $x, y, z$ ?
- ¿Con cuales elementos está conectado de la posición  $x, y, z$ ?

2. **Conceptualización.** En esta etapa se elabora el modelo conceptual, para lo cual se deben desarrollar una serie de tareas que se muestran en la Figura 5.2.5.

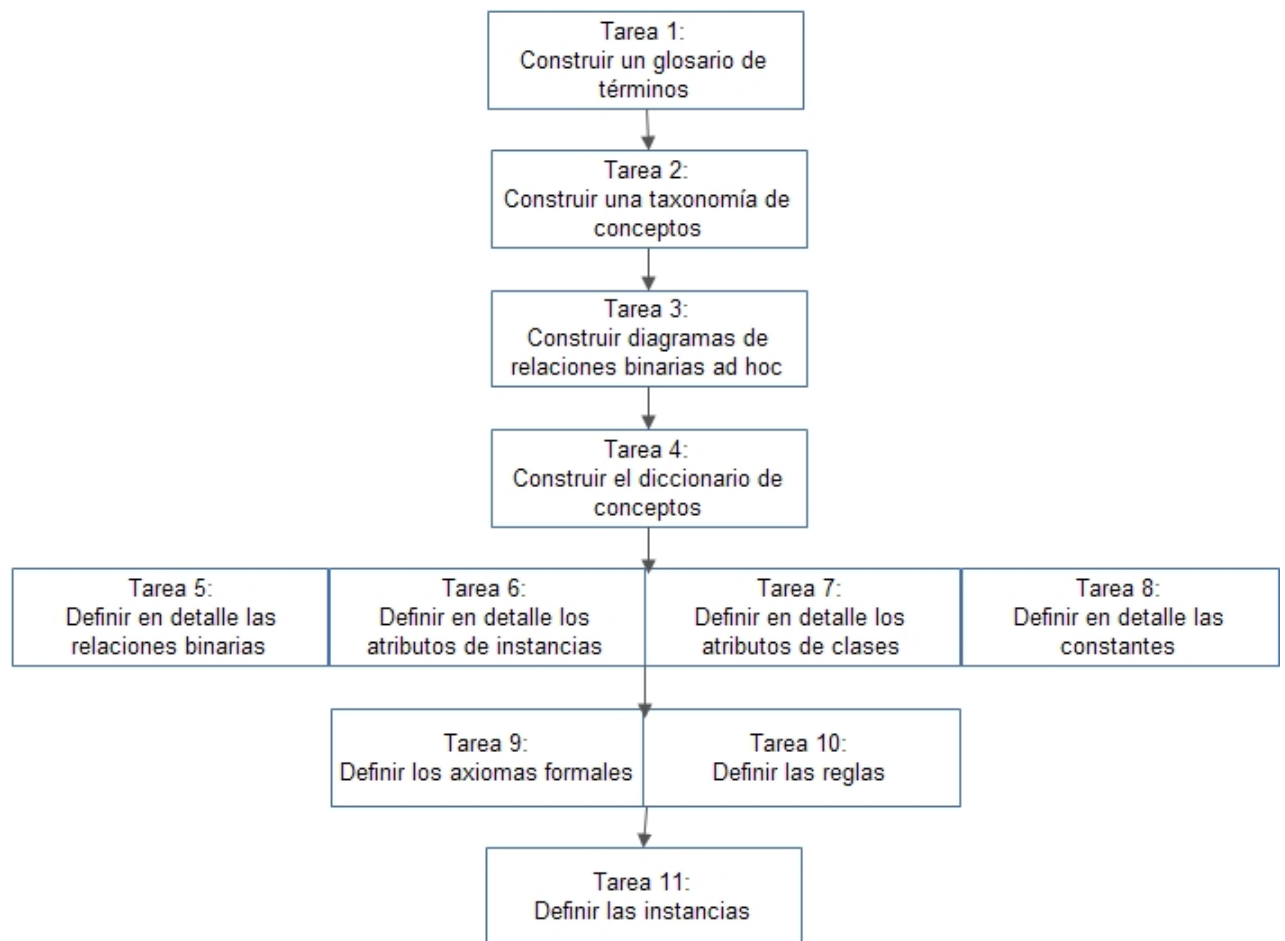


Fig. 5.2.5: Tareas de la etapa de Conceptualización de acuerdo con Methontology

- Tarea 1: Construir un glosario de términos. Esta tarea tiene el objetivo de identificar y describir los conceptos fundamentales del dominio. La Tabla 5.2.7 muestra la definición de los conceptos de la ontología propuesta en esta tesis.

Tabla 5.2.7: Glosario de términos

<i>Nombre</i>	<i>Sinónimo</i>	<i>Descripción</i>	<i>Tipo</i>
PointCloud		El objeto <i>PointCloud</i> representa un conjunto de puntos en un sistema de coordenadas tridimensional, que describen una escena del mundo real.	Concepto
Centroid		El objeto <i>Centroid</i> describe un punto en el sistema de coordenadas tridimensional, que se encuentra justo en el centro de un objeto	Concepto
Voxel		Un <i>Voxel</i> representa una cuadrícula o rejilla en el espacio 3D en la nube de puntos; un objeto <i>Voxel</i> delimita a un <i>Shape</i> bajo límites espaciales.	Concepto
Shape		Un <i>Shape</i> es un objeto geométrico tridimensional que ocupa un lugar en el espacio: se puede clasificar como <i>Surface</i> o <i>Volumetric</i> . Cada elemento <i>Shape</i> pertenece a un <i>PointCloud</i> específico.	Concepto
SpatialRelation		Un <i>SpatialRelation</i> describe las interacciones espaciales entre <i>Referent</i> y el <i>Relatum</i>	Concepto
Referent		Un <i>Referent</i> es un <i>Shape</i> que describe el objetivo relativo a partir del que se describe la relación espacial.	Relación
Relatum		A <i>Relatum</i> es un un <i>Shape</i> , hace referencia a un objeto con respecto al cual el <i>Referent</i> va a ser ubicado.	Relación
SpatialRepresentation		Define la representación espacial entre el <i>Referent</i> y <i>Relatum</i> .	Relación
GlobalRelationship		Define la relación global entre el <i>Referent</i> y el <i>Relatum</i>	Relación
Adjacency		Un <i>Adjancecy</i> describe la cualidad de un objeto a estar suficientemente próximo a fin de tocar a otro objeto.	Concepto
Continúa en la siguiente página			

Tabla 5.2.7 – continuación de página previa

Nombre	Sinónimo	Descripción	Tipo
SideAdjacency		Un <i>SideAdjacency</i> describe la relación de estar al lado de otro elemento dentro de la escena considerando uno de los lados del objeto ( <i>Right</i> , <i>Left</i> , <i>Front</i> , <i>Back</i> , <i>Above</i> y <i>Below</i> )	Concepto
TotalAdjacency		Un <i>TotalAdjacency</i> describe la relación de estar cercano de otro elemento dentro de la escena considerando el total de los puntos del objeto.	Concepto
Side		Un <i>Side</i> describe el costado de un objeto, es decir las líneas que forman una figura. Un <i>Side</i> puede tener los valores <i>Right</i> , <i>Left</i> , <i>Front</i> , <i>Back</i> , <i>Above</i> y <i>Below</i>	Concepto
Surface		Un <i>Surface</i> es un tipo de <i>Shape</i> , que describe un objeto que tiene longitud y anchura.	Concepto
Plane		Un <i>Plane</i> es un <i>Surface</i> , que se compone de puntos que tienen la misma orientación dentro del plano. En el mundo real, es fácil de encontrar objetos <i>Plane</i> tales como mesas, techos, paredes, aceras, calles y hojas de papel, por citar algunos ejemplos.	Concepto
Volumetric		Un elemento <i>Volumetric</i> es un tipo de <i>Shape</i> , que describe una forma geométrica tridimensional que tiene volumen.	Concepto
Sphere		El objeto <i>Sphere</i> describe todos los elementos que tienen una superficie esférica. Un <i>Sphere</i> se pueden encontrar tanto en entornos artificiales como en la naturaleza; por ejemplo, una pelota, un limón, una perilla de la puerta y una bombilla, por nombrar algunos.	Concepto
Continúa en la siguiente página			

Tabla 5.2.7 – continuación de página previa

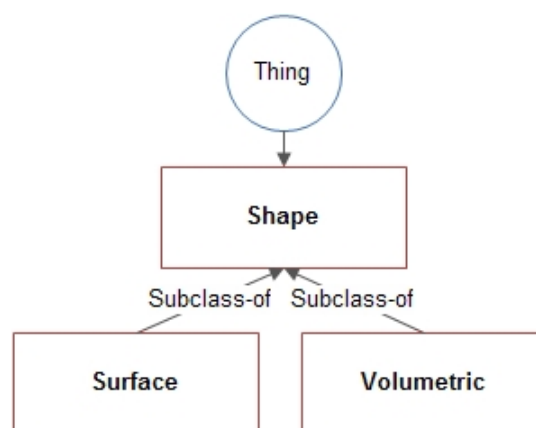
Nombre	Sinónimo	Descripción	Tipo
Torus		Un <i>Torus</i> es un elemento de la ontología que describe un pequeño círculo que gira a lo largo de la línea trazada por el otro círculo. Muchos de los objetos de la vida real tienen un toro: Una dona, un cojín romano, un neumático, una bombilla, entre otros.	Concepto
Cylinder	Pipe	Un objeto <i>Cylinder</i> describe elementos formados por curvas simples creadas a través del movimiento de una línea recta alrededor de la circunferencia de un círculo. Es posible encontrar un cilindro en el mundo real en la forma de botellas, latas de refrescos, columnas, tuberías, entre otras.	Concepto
Wall		Un <i>Wall</i> es un <i>Plane</i> y define una partición arquitectónica con una altura y una longitud; utilizado para dividir o encerrar un área.	Concepto
Tee		El <i>Tee</i> es un <i>Sphere</i> , el cual describe al conector más común de una tubería, se utiliza para combinar o bien dividir un flujo de fluidos. Es un conector de tuberías con forma de <i>T</i> con dos salidas, a 90° a la conexión de la línea principal	Concepto
Valve		Un <i>Valve</i> es un <i>Torus</i> que representa un conector de tubería diseñado para detener o regular el flujo de cualquier fluido que pase a través de ella.	Concepto
Elbow		El <i>Elbow</i> es un <i>Torus</i> . Este elemento describe un accesorio real de tubería instalada entre dos longitudes de tubo, el cual permite un cambio de dirección del flujo, por lo general un ángulo de 90° o 45°.	Concepto
Pipe		Un <i>Pipe</i> es un <i>Cylinder</i> , el cual describe una sección cilíndrica utilizada para transportar sustancias que pueden fluir, suspensiones, polvos o masas de sólidos.	Concepto
Continúa en la siguiente página			



Tabla 5.2.7 – continuación de página previa

Nombre	Sinónimo	Descripción	Tipo
Manufacturer		Un elemento <i>Manufacturer</i> , representa una empresa dedicada a la fabricación de algún producto específico.	Concepto

- Tarea 2: Construir una taxonomía de conceptos. Una vez obtenido el glosario, se debe seleccionar aquellos términos que son conceptos y crear una taxonomía, la cual define la jerarquía entre los conceptos del dominio. La Figura 5.2.6 muestra la definición formal de la taxonomía del elemento *Shape* el cual puede ser clasificado como *Surface* o *Volumetric*. Por otra parte, la Figura 5.2.7 muestra la definición formal de la taxonomía del elemento *Surface* el cual tiene relacionada la clase *Plane*. Mientras que la Figura 5.2.8 presenta la definición formal de la taxonomía del elemento *Volumetric*, un elemento *Volumetric* puede ser de tipo *Sphere*, *Torus* o *Cylinder*. La Figura 5.2.9 presenta el concepto *Adjacency* el cual se puede clasificar como *Side Adjacency* o *Total Adjacency*, al referirse a una relación de proximidad por una de sus caras o total del objeto con otro objeto dentro de la escena industrial. Finalmente la Figura 5.2.10 muestra el concepto *Manufacturer* el cual se puede clasificar como *ElbowManufacturer*, *ValveManufacturer*, *PipeManufacturer* o *TeeManufacturer*, que representa a los productores de Codos, Válvulas, Tubos y Tes.

Fig. 5.2.6: Taxonomía general de *Shape*

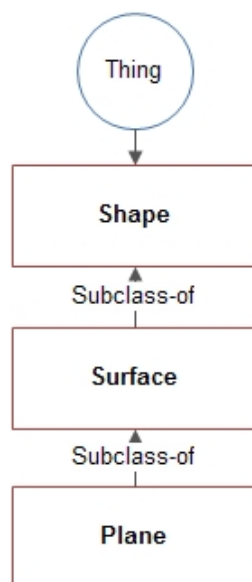


Fig. 5.2.7: Taxonomía *Surface*

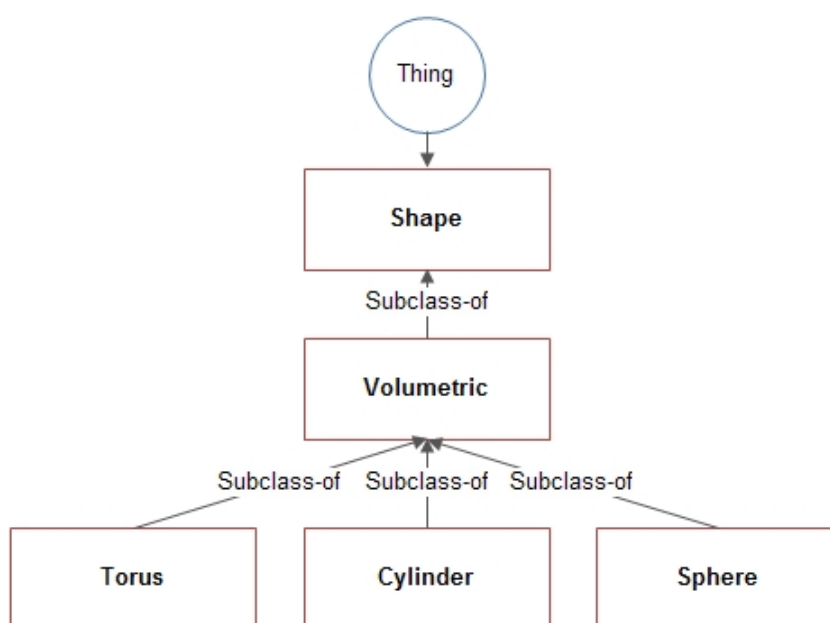
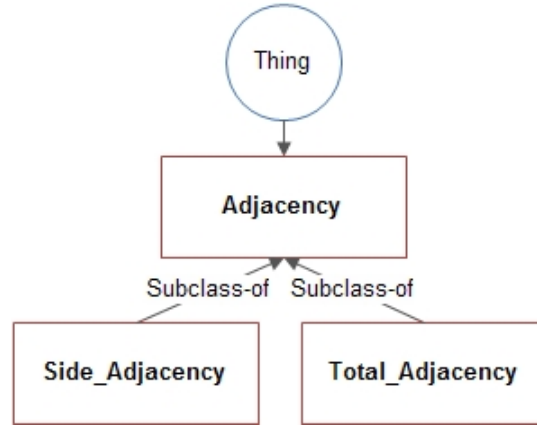
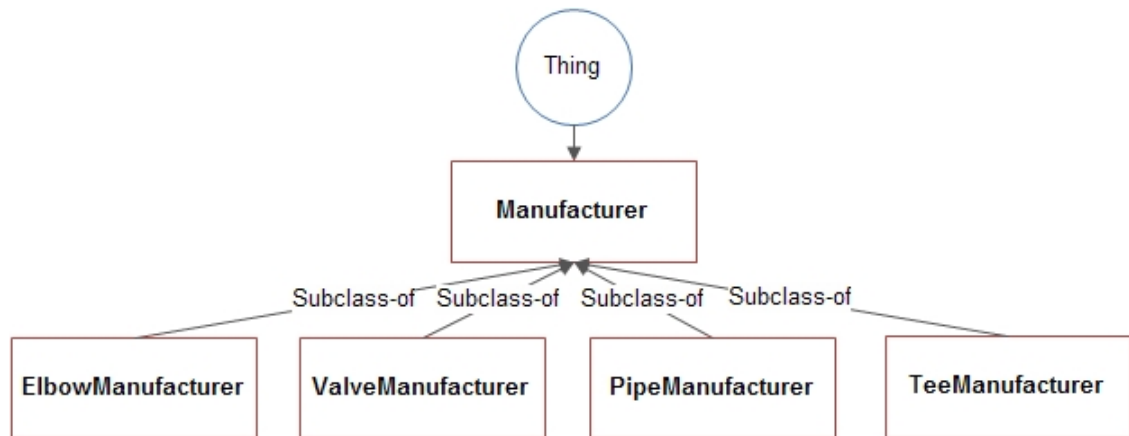


Fig. 5.2.8: Taxonomía *Volumetric*

Fig. 5.2.9: Taxonomía *Adjacency*Fig. 5.2.10: Taxonomía *Manufacturer*

- Tarea 3: Construir un diagrama de relaciones binarias. Ad hoc es una locución del latín que significa “para esto”, es decir, creado o hecho para un fin determinado, según sea necesario. Los diagramas Ad hoc especifican las relaciones entre los conceptos y clases del dominio. La Figura 5.2.11 muestra los diagramas de relación binaria Ad hoc, donde se presentan las siguientes relaciones:
  - Un *PointCloud* tiene algunos(*has\_some*) *Shape*, mientras que un *Shape* es parte de (*is\_part\_of*) un *Pointcloud*. Es decir, que una nube de puntos *PointCloud* está conformada por un conjunto de *Shapes*.
  - Un *Shape* tiene un(*has\_a*) *Centroid*, mientras que, en la relación inversa, un *Centroid* es pertenece a un (*belong\_to*) un *Shape*. Esto es, que un *Shape*

tinene asociado un *Centroid* que indica el punto central de la forma en el espacio.

- Un *Shape* tiene un(*has\_a*) *Voxel*, y la relación inversa, un *Voxel* es pertenece a un (*belong\_to*) un *Shape*. Esto significa que un *Shape* tinene asociado un *Voxel*, un cuadrado que enmarca a la forma en el espacio.
  - Una *SpatialRelation* es una relación que pertenece a (*belong\_to*) un solo *Shape*, esta relación tiene asociados dos elementos *Shape* los cuales actúan como relatum y referent (*has\_a\_Relatum* y *has\_a\_Referent*), de manera inversa, cada *Shape* tiene una *SpatialRelation*.
  - Un *SpatialRelation* tiene algunas (*has\_some*) *Adjacency*, pero cada *Adjacency* tiene solo una *SpatialRelation* asociada.
  - Cada *Adjacency* puede clasificarse (*is\_classified*) en *Side\_Adjacency* o *Total\_Adjacency*.
  - Cada *Adjacency* hace referencia (*reference\_to\_side*) a un lado del *Shape* asociado.
  - Un *Shape* puede ser clasificado (*is\_classified\_as*) como *Volumetric* para formas con una representación visual en tres dimensiones físicas, o *Surface* para el conjunto de puntos de un espacio euclídeo que forma un espacio topológico bidimensional.
  - Un *Surface* puede ser clasificada (*is\_classified\_as*) como *Plane*.
  - Un *Volumetric* puede ser clasificado (*is\_classified\_as*) como *Cylinder*, *Sphere* o *Torus*.
  - En el contexto industrial, un *Cylinder* es clasificado (*is\_classified\_as*) como *Pipe*, un *Torus* puede ser clasificado (*is\_classified\_as*) como un *Elbow* o una *Valve* y un *Sphere* puede ser clasificado (*is\_classified\_as*) como un *Tee*.
  - En el contexto industrial, un *Cylinder* es clasificado (*is\_classified\_as*) como *Pipe*, un *Torus* puede ser clasificado (*is\_classified\_as*) como un *Elbow* o una *Valve* y un *Sphere* puede ser clasificado (*is\_classified\_as*) como un *Tee*.
  - Cada elemento tiene asociados algunos *Manufacturer*. Y los *Manufacturer* son clasificados (*is\_classified\_as*) como *PipeManufacturer*, *ElbowManufacturer*, *TeeManufacturer* o *ValveManufacturer*.
- Tarea 4: Construir el diccionario de conceptos. En esta etapa se define un diccionario de conceptos que incluye En este diccionario se incluyen todos los conceptos del vocabulario, sus relaciones binarias y propiedades a nivel de concepto y a nivel de instancia. La Tabla 5.2.8 muestra el diccionario de conceptos elaborado para esta tesis.



Fig. 5.2.11: Diagrama de Relaciones Binarias Ad hoc

Tabla 5.2.8: Diccionario de conceptos

Concepto	Atributo de clase	Atributo de instancia	Relaciones
PointCloud	numberOfPoints	name	has_some
Centroid		x, y, y z	belong_to
Voxel		x1, y1, z1, x2, y2, z2, x3, y3, z3, x4, y4, z4 x5, y5, z5, x6, y6, z6, x7, y7, z7, x8, y8, z8	belong_to
Shape	numberOfPoints	name	has_a
SpatialRelation		Referent Relatum SpatialRepresentation GlobalRepresentation	belong_to has_a_Relatum has_a Referent
Surface		length width	is_a
Adjacency		isValid percentage	is_a reference_to_side
Side_Adjacency			is_classified_as
Total_Adjacency			is_classified_as
Side			-
Plane		name numberOfPoints length width	is_a
Volumetric	numberOfPoints	name length	is_a is_classified_as
Sphere		name numberOfPoints length height radius diameter	is_a is_classified_as

Continúa en la siguiente página

Tabla 5.2.8 – continuación de página previa

Concepto	Atributo de clase	Atributo de instancia	Relaciones
Torus		name numberOfPoints length minorRadius majorRadius	is_a is_classified_as
Cylinder		name numberOfPoints length height radius diameter	is_a is_classified_as
Pipe		numberOfPoints length height radius diameter nominalDiameter	is_a has_some
Tee		nominalSize	is_a has_some
Elbow		nominalSize	is_a has_some
Valve		nominalSize	is_a has_some
Manufacturer		name	is_a
ValveManufacturer		technicalSpecification dn_mm pn_mpa pipelineType standard stainlessSteel nominalSize d l h1 h2 weight	is_a

Continúa en la siguiente página

Tabla 5.2.8 – continuación de página previa

Concepto	Atributo de clase	Atributo de instancia	Relaciones
ElbowManufacturer		nominalSize	is_a
		a	
		weightBlack	
		weightGalv	
		nominalDiameter	
PipeManufacturer		certifiedManufacturing	is_a
		externalDiameter	
		manufacturerMaterial	
		description	
		finishingTouch	
		thickness	
		Model	
TeeManufacturer		theoreticalWeight	is_a
		supply	
		quality	
		nominalSize	
		model	
		weightBlack	
		weightGalv	
		WSP	
		WOG	
		quality	
		a_mm	
		b_mm	
		c_mm	
		a_kg	
		b_kg	
		c_kg	
		thruHole	

- Tarea 5: Definir en detalle las relaciones binarias. Esta etapa describirá para cada relación binaria nombre, nombres de sus conceptos origen y destino, cardinalidad, relación inversa, restricciones del tipo de valores que se aceptaran si existe. La Tabla 5.2.9 muestra el detalle de las relaciones binarias.



Tabla 5.2.9: Relaciones binarias detalladas

<i>Nombre de la relación</i>	<i>Concepto origen</i>	<i>Concepto destino</i>	<i>Cardinalidad</i>	<i>Relación inversa</i>
has_some	PointCloud	Shape	1..N	is_part_of
has_some	SpatialRelation	Adjacency	1..6	has_a
has_some	Elbow	ElbowManufacturer	1..N	-
has_some	Tee	TeeManufacturer	1..N	-
has_some	Valve	ValveManufacturer	1..N	-
has_some	Pipe	PipeManufacturer	1..N	-
has_a	Adjacency	SpatialRelation	1,1	has_some
has_a	Shape	Centroid	1,1	belong_to
has_a	Shape	SpatialRelation	1,1	belong_to
has_a	Shape	Voxel	1,1	belong_to
has_a_Relatum	SpatialRelation	Shape	0,1	is_related_with
has_a_Referent	SpatialRelation	Shape	0,1	is_related_with
is_a	Adjacency	Side_Adjacency	0,1	is_classified_as
is_a	Adjacency	Total_Adjacency	0,1	is_classified_as
is_a	Shape	Volumetric	0,1	is_classified_as
is_a	Shape	Suface	0,1	is_classified_as
is_a	Surface	Plane	0,1	is_classified_as
is_a	Volumetric	Cylinder	0,1	is_classified_as
is_a	Volumetric	Torus	0,1	is_classified_as
is_a	Volumetric	Sphere	0,1	is_classified_as
is_a	Sphere	Tee	0,1	is_classified_as
is_a	Cylinder	Pipe	0,1	is_classified_as
is_a	Torus	Elbow	0,1	is_classified_as
is_a	Torus	Valve	0,1	is_classified_as
is_a	ElbowManufacturer	Manufacturer	1,1	is_classified_as
is_a	ValveManufacturer	Manufacturer	1,1	is_classified_as
is_a	PipeManufacturer	Manufacturer	1,1	is_classified_as
is_a	TeeManufacturer	Manufacturer	1,1	is_classified_as

- Tarea 6: Describir en detalle los atributos de instancias. Se especificarán las características de los atributos de las instancias. Los atributos de instancias son aquellos cuyos valores pueden ser diferentes para cada instancia del concepto. Se detallan considerando su nombre, concepto al que pertenece, tipo de valor, rango de valores y cardinalidad. La Tabla 5.2.10 muestra el destalle de los atributos de las instancias.

Tabla 5.2.10: Detalle de atributos de instancia

<i>Nombre del atributo de la instancia</i>	<i>Concepto</i>	<i>Tipo de valor</i>	<i>Rango de valores</i>	<i>Cardinalidad</i>
name	PointCloud			
	Shape			
	Surface	Cadena de caracteres	-	1,1
	Volumetric			
x	Centroid	Flotante	1.18e-38 - 3.40e38	1,1
y	Centroid	Flotante	1.18e-38 - 3.40e38	1,1
z	Centroid	Flotante	1.18e-38 - 3.40e38	1,1
x1, x2, x3,x4	Voxel	Flotante	1.18e-38 - 3.40e38	1,1
y1, y2, y3, y4	Voxel	Flotante	1.18e-38 - 3.40e38	1,1
z1, z2, z3, z4	Voxel	Flotante	1.18e-38 - 3.40e38	1,1
Referent	SpatialRelation	Shape	-	0,1
Relatum	SpatialRelation	Shape	-	0,1
isValid	Adjacency	Booleano	True-False	1,1
percentage	Adjacency	Flotante	1.18e-38 - 3.40e38	1,1
SpatialRepresntation	SpatialRelation	Cadena de caracteres	-	0,1
GlobalRelationship	SpatialRelation	Cadena de caracteres	-	0,1
legth	Surface			
	Volumetric	Flotante	1.18e-38 - 3.40e38	1,1
width	Surface	Flotante	1.18e-38 - 3.40e38	1,1
height	Sphere			
	Cylinder	Flotante	1.18e-38 - 3.40e38	1,1
radius	Sphere			
	Cylinder	Flotante	1.18e-38 - 3.40e38	1,1
diameter	Sphere			
	Cylinder	Flotante	1.18e-38 - 3.40e38	1,1
minorRadius	Torus	Flotante	1.18e-38 - 3.40e38	1,1
majorRadius	Torus	Flotante	1.18e-38 - 3.40e38	1,1
nominalDiameter	Pipe			
	PipeManufacturer	Flotante	1.18e-38 - 3.40e38	1,1
certiedManufacturing	PipeManufacturer	Cadena de caracteres	-	0,1
externalDiameter	PipeManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
manufacturerMaterial	PipeManufacturer	Cadena de caracteres	-	0,1
description	PipeManufacturer	Cadena de caracteres	-	0,1
finishingTouch	PipeManufacturer	Cadena de caracteres	-	0,1
thicknessModel	PipeManufacturer	Cadena de caracteres	-	0,1
Continúa en la siguiente página				

Tabla 5.2.10 – continuación de página previa

<i>Nombre del atributo de la instancia</i>	<i>Concepto</i>	<i>Tipo de valor</i>	<i>Rango de valores</i>	<i>Cardinalidad</i>
theoreticalWeight	PipeManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
supply	PipeManufacturer	Cadena de caracteres	-	0,1
quality	PipeManufacturer	Cadena de caracteres	-	0,1
nominalSize	Tee			
	Elbow			
	Valve			
	ValveManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
	ElbowManufacturer			
model	TeeManufacturer	Cadena de caracteres	-	0,1
	ElbowManufacturer			
weightBlack	TeeManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
weightGalv	ElbowManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
WSP	TeeManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
WOG	TeeManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
quality	TeeManufacturer	Cadena de caracteres	-	0,1
a_mm	ElbowManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
b_mm	TeeManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
c_mm	TeeManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
a_kg	TeeManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
b_kg	TeeManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
c_kg	TeeManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
thruHole	TeeManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
technicalSpecication	ValveManufacturer	Cadena de caracteres	-	0,1
dn_mm	ValveManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
pn_mpa	ValveManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
pipelineType	ValveManufacturer	Cadena de caracteres	-	0,1
standard	ValveManufacturer	Cadena de caracteres	-	0,1
stainlessSteel	ValveManufacturer	Cadena de caracteres	-	0,1
d	ValveManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
l	ValveManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
h1	ValveManufacturer	Flotante	1.18e-38 - 3.40e38	0,1
h2	ValveManufacturer	Flotante	1.18e-38 - 3.40e38	0,1



- Tarea 7: Describir en detalle los atributos de clase. En esta etapa se especifica las características de los atributos de la clase. Los atributos de clases representan características genéricas de un concepto, es decir todas las instancias del concepto tendrán el mismo valor para ese atributo. La descripción consta del nombre del atributo, nombre del concepto, tipo de valor, rango de valores y cardinalidad. La Tabla 5.2.11 presenta el detalle de los atributos de la clase.

Tabla 5.2.11: Atributos de la clase

<i>Nombre del atributo de la clase</i>	<i>Concepto</i>	<i>Tipo de valor</i>	<i>Rango de valores</i>	<i>Cardinalidad</i>
numberOfPoints	PointCloud	entero	1..	1,1
	Shape			
	Surface			
	Volumetric			

A su vez, la Tabla 5.2.12 muestra una breve descripción de cada uno de los atributos.

Tabla 5.2.12: Descripción de atributos

<i>Atributo</i>	<i>Definición</i>	<i>Concepto</i>
Name	Identifica el elemento con el fin de referirse an él	PointCloud, Plane, Volumetric, Sphere, Torus, Cylinder, Pipe
NumberOfPoints	Número total de puntos que constituyen el elemento	PointCloud, Plane, Volumetric, Sphere, Torus, Cylinder, Pipe
x, y, z	Coordenadas de un punto en el espacio euclidiano	Centroid
x1, y1, z1	Describen las coordenadas de las ocho esquinas de un voxel. X1, y1 y z1 representa la coordenada de la vista frontal de la,esquina superior izquierda del voxel.	Voxel
Continúa en la siguiente página		

Tabla 5.2.12 – continuación de página previa

<i>Atributo</i>	<i>Definición</i>	<i>Concepto</i>
x2, y2, z2	Representa la coordenada de la vista frontal de la, esquina superior derecha del voxel.	Voxel
x3, y3, z3	Representa la coordenada de la vista frontal de la,esquina inferior izquierda del voxel.	Voxel
x4, y4, z4	Representa la coordenada de la vista frontal de la,esquina inferior derecha del voxel.	Voxel
x5, y5, z5	Representa la coordenada de la vista posterior de,la,esquina superior izquierda del,voxel.	Voxel
x6, y6, z6	Representa la coordenada de la vista posterior de la, esquina superior derecha del voxel.	Voxel
x7, y7, z7	Representa la coordenada de la vista posterior de la,esquina inferior izquierda del voxel.	Voxel
x8, y8, z8	Representa la coordenada de la vista posterior de la, esquina inferior derecha del voxel.	Voxel
Referent	Describe el objetivo,relativo a partir del que se describe la relación espacial.	SpatialRelation
Relatum	Hace referencia a un,objeto con respecto al cual el Referent va a ser ubicado.	SpatialRelation
SpatialRepresentation	Define la representación espacial entre el Referent y Relatum.	SpatialRelation
GlobalRepresentation	Define la relación global entre el Referent y el Relatum	SpatialRelation
length	Representa la extensión lineal en el espacio de un elemento de un,extremo al otro	Surface, Plane, Volumetric, Sphere, Torus, Cylinder, Pipe
width	Medida de un elemento de lado a lado	Surface, Plane
Continúa en la siguiente página		

Tabla 5.2.12 – continuación de página previa

<i>Atributo</i>	<i>Definición</i>	<i>Concepto</i>
isValid	Marca a un elemento fundamentado correctamente en la lógica o la verdad	Adjacency, Side_Adjacency, Total_Adjacency
percentage	Proporción en relación con un conjunto	Adjacency, Side_Adjacency, Total_Adjacency
radius	Longitud de un segmento de línea que va desde el centro y la circunferencia de un círculo o esfera	Sphere, Cylinder, Pipe
minorRadius	Representa el diámetro más corto dentro de un toroide	Torus
majorRadius	Representa el diámetro más grande dentro de un toroide	Torus
height	Distancia de la base del elemento a la parte superior	Sphere, Pipe
NominalSize	El tamaño nominal puede corresponder a un gran número de dimensiones y tolerancias muy estandarizados y de acuerdo al elemento. Son utilizados para la identificación	Elbow, Tee, Valve, ValveManufacturer, ElbowManufacturer, TeeManufacturer
a	Representa el radio largo, que va del centro del elemento al final	ElbowManufacturer
Diameter	Longitud de una línea recta que pasa a través del centro de un círculo y que conecta dos puntos en la circunferencia	Sphere, Cylinder
nominalDiameter	El diámetro nominal de tubos representa el tamaño estándar para tuberías de presión. En Estados Unidos se usa un sistema denominado en pulgadas, mientras en Europa denominan en milímetros según lo define la norma ISO 6708	Pipe, PipeManufacturer
certifiedManufacturing	Se refiere al certificado de manufactura con el que fue elaborado el elemento	PipeManufacturer
externalDiameter	Diámetro del círculo exterior.	PipeManufacturer
manufacturerMaterial	Representa al nombre del posible fabricante	PipeManufacturer
Continúa en la siguiente página		

Tabla 5.2.12 – continuación de página previa

<i>Atributo</i>	<i>Definición</i>	<i>Concepto</i>
description	Descripción breve del tubo	PipeManufacturer, ValveManufacturer, TeeManufacturer
finishingTouch	Detalle de distinción, detalle de terminación	PipeManufacturer, TeeManufacturer, ValveManufacturer
thicknessModel	Define el grosor del tubo	PipeManufacturer
theoreticalWeight	Representa el peso teórico del tubo de acuerdo al estándar de fabricación	PipeManufacturer
quality	Especifica el estandar de calidad del tubo	PipeManufacturer, TeeManufacturer, ValveManufacturer
model	Define el modelo del elemento	TeeManufacturer
Weight	La fuerza vertical ejercida por una masa como un resultado de la gravedad	TeeManufacturer, ElbowManufacturer, ValveManufacturer
WSP	Define la presión del vapor de agua o la presión de vapor de trabajo. Refleja la presión de la válvula en la temperatura del vapor de agua.	TeeManufacturer
WOG	WOG (Water - Oil - Gas) Usado en conjunto con la clasificación de presión, así: 100 WOG indica la presión nominal es de 100 psi en agua, aceite o el servicio de gas, a temperaturas ambientales normales.	TeeManufacturer
a mm	Diámetro exterior	TeeManufacturer
b mm	Diámetro interior al Final	TeeManufacturer
type	Especifica el tipo de válvula: Gates, Globes, Angles, Venthead, Storm, Fire Hose, Ball, Checks (Wafer, Duo, Disc, Lift and Swing)	ValveManufacturer
PressureClass	Describe la clase de presión soportada	ValveManufacturer
Materials	Materiales con que fue construida	ValveManufacturer

- Tarea 8: Definir en detalle las constantes. Esta tarea no aplica a este dominio debido a que no se presentan constantes.
  - Tarea 9: Definir los axiomas formales. El detalle de la definición de los axiomas se describe de manera más extensa en la sección 5.2.2.5.
  - Tarea 10: Definir las reglas. La descripción detallada de las reglas se presenta de manera más amplia en la sección 5.2.2.5.
  - Tarea 11: Describir instancias. En esta etapa se deben presentar a detalle instancias si éstas son conocidas. Para esta tesis esta etapa no aplica.
3. Implementación. Esta etapa conlleva el aplicar todo lo analizado en pasos anteriores de manera que la ontología resultante refleje la totalidad de los detalles. En el caso de esta tesis, la ontología se desarrolló con Protégé <sup>6</sup>, Protégé es una herramienta *Standalone* libre desarrollado en la Universidad de Stanford e implementado en java, el cual permite la construcción de ontologías (N. F. Noy et al., 2000). Protégé fue desarrollada inicialmente por Mark Musen en 1987, la versión original fue una pequeña aplicación, destinada a la construcción de herramientas para la adquisición de conocimiento por parte de algunos programas con enfoques médicos. Desde entonces Protégé ha evolucionado hasta ser una herramienta multi-plataforma extensible para el desarrollo de sistemas basados en conocimiento e investigación. En su versión actual, conserva el editor de ontologías como su núcleo, soporta extensiones de interfaz de usuario personalizadas, ofrece una biblioteca de plug-ins que le añaden funcionalidad, además de permitir la exportación de ontologías hacia otros lenguajes entre los cuales están F-Logic, OIL y XML, así como hacia motores de inferencia como Jess y Prolog (Gennari et al., 2003). Protégé ofrece una interfaz gráfica, la cual permite al usuario enfocarse en el modelado conceptual, sin que este requiera de conocimientos de sintaxis de lenguajes de salida. El modelo de conocimiento de Protégé está basado en frames, por lo que sus elementos proporcionan clases, propiedades e instancias de las mismas.

La implementación con Protégé del modelo conceptual generado para esta tesis, se realizó de la siguiente manera:

- Definición de clases especificando jerarquías.
  - Definición de las relaciones entre clases, considerando atributos, dominios, tipos y cardinalidad detalladas a través de la metodología desarrollada.
4. Evaluación. Dentro de esta etapa se comprueba el correcto funcionamiento de la ontología mediante pruebas verificando posibles inconsistencias y redundancias. El detalle de la evaluación se describe de manera más extensa en el capítulo 5.

---

<sup>6</sup><http://protege.stanford.edu/>



### 5.2.2.2 Resultados de la Ontología propuesta

La Figura 5.2.12 muestra la definición formal de primitivas geométricas, las cuales tiene la capacidad de estar asociados con otras ontologías externas como DBpedia, estándares de calidad de productos o fabricación u ontologías privadas, entre las que podemos mencionar ontologías de elementos de fabricación, artículos de inventario, mobiliario de oficina, productos comerciales y fabricantes, para nombrar unos pocos.

### 5.2.2.3 Cálculo de características geométricas

Este sub-proceso tiene como objetivo para calcular propiedades geométricas específicas de acuerdo con el tipo de primitiva geométrica reconocida, como su longitud, diámetro, altura y radio.

La Tabla 5.2.13 muestra las propiedades geométricas a calcular considerando el tipo de primitiva. Después, toda la información calculada se almacena, considerando la definición de la ontología propuesta.

Tabla 5.2.13: Propiedades geométricas de acuerdo a la primitiva

Propiedades geométricas	<i>PointCloud</i>	<i>Surface</i>	<i>Plane</i>	<i>Shape</i>	<i>Cylinder</i>	<i>Torus</i>	<i>Sphere</i>	<i>Cone</i>
NumberOfPoints	x	x	x	x	x	x	x	x
Length		x	x	x	x	x	x	x
Width		x	x					
Height					x		x	x
Radius					x		x	x
Diameter					x		x	x
MinorRadius						x		
MajorRadius						x		

### 5.2.2.4 Resultado del Cálculo de características geométricas

De acuerdo a lo especificado en la Tabla 5.2.13. La Figura 5.2.13 muestra una consulta de las propiedades calculadas para un tubo.

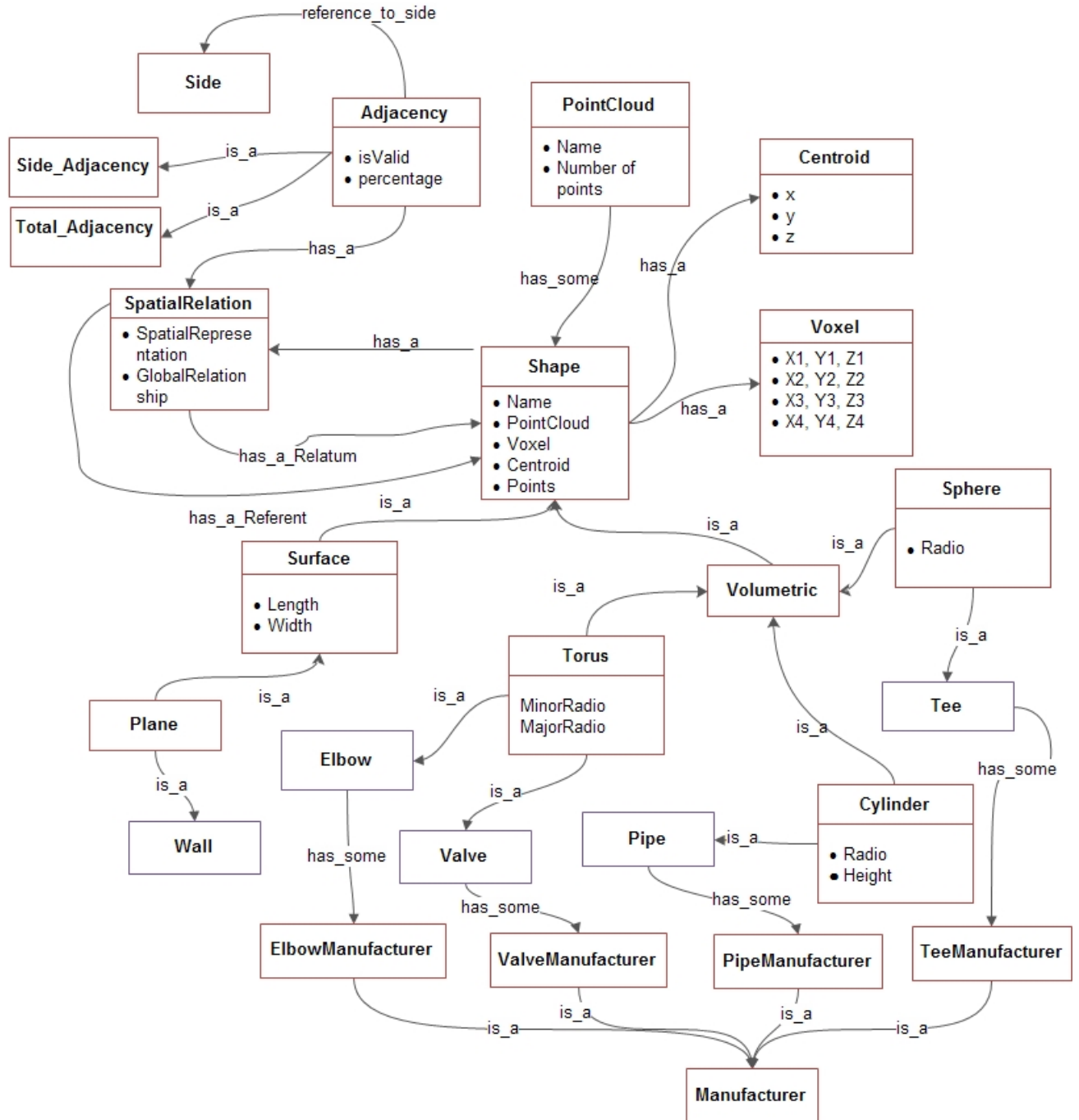


Fig. 5.2.12: Ontología propuesta

SPARQL query:	
<pre> PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX owl: &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt; prefix ont: &lt;http://nadir.uc3m.es/yphd/ont.owl#&gt; SELECT * WHERE {   ont:PIPE_002 ?properties ?value .   FILTER isLiteral(?value) } </pre>	
properties	value
numberOfPoints	"20142"^^<http://www.w3.org/2001/XMLSchema#integer>
length	"3.656349"^^<http://www.w3.org/2001/XMLSchema#decimal>
radius	"0.167465"^^<http://www.w3.org/2001/XMLSchema#decimal>
diameter	"0.33493"^^<http://www.w3.org/2001/XMLSchema#decimal>
nominalDiameter	"152.4"^^<http://www.w3.org/2001/XMLSchema#decimal>

Fig. 5.2.13: Propiedades geométricas de un tubo

Toda la información calculada se almacena, teniendo en cuenta la definición de la ontología propuesta. En este punto, el algoritmo tiene una asociación lógica de elementos segmentados de la nube de puntos, además de un significado semántico para cada uno. El siguiente paso del algoritmo es calcular las relaciones espaciales entre los objetos pertenecientes a la nube de puntos.

#### 5.2.2.5 Cálculo de la topología: relaciones espaciales

Hay dos tipos de relaciones espaciales que definen la topología en el presente algoritmo: representación espacial y la relación espacial. Considerando el contexto del caso de uso en el que se valida la presente tesis, es decir, el análisis de los escenarios industriales, la

Tabla 5.2.14 muestra las relaciones posibles para cada objeto que será reconocido. Como se ha señalado, estos elementos se han limitado a tuberías, codos, tes y válvulas, por lo que no se permiten todas las relaciones globales. Fueron definidos el número de posibles relaciones globales para cada objeto *Referent* de acuerdo con sus características reales y el o los elementos *Relatum* esperados. Por ejemplo, un tubo puede tener sólo dos conexiones, que pueden ser codos o tes; y una válvula tiene sólo una conexión porque en el mundo real, las válvulas están por lo general sólo conectados a una te.

Tabla 5.2.14: Relaciones Globales posibles

<i>Referent</i>	<i>Núm de posibles relaciones</i>	<i>Relatum esperado</i>
Pipe	2	Elbow Tee
Elbow	2	Pipe
Tee	3	2 Pipes y 1 Valve 3 pipes
Valve	1	Tee

El algoritmo valora estas condiciones a partir de la definición e implementación de reglas semánticas. El condicional del número de relaciones posibles para cada elemento se solventa a partir de afirmaciones semánticas como las que se describen abajo.

**Afirmaciones:**

```

is_a(pipe, cylinder)      assert-1
is_a(elbow, torus)        assert-2
is_a(valve, torus)        assert-3
is_a(tee, sphere)         assert-4
num_of_connections(pipe, 2)  assert-5
num_of_connections(elbow, 2) assert-6
num_of_connections(valve, 1) assert-7
num_of_connections(tee, 3)  assert-8

```

Por otra parte, el condicional que describe las relaciones espaciales permitidas por el algoritmo se solventa con un conjunto de reglas semánticas denominadas *Spatial Relationship Rules*. La lista de las Spatial Relationship Rules creadas se presenta a continuación, para cada una se muestra la definición formal y la respectiva regla en sintaxis de Jena. Un ejemplo de este tipo de reglas se muestra gráficamente en la Figura 5.2.14

- Una conexión tipo **A Left B** es válida si, y sólo si, los puntos de la izquierda del Shape A están 90% cerca de los puntos derechos del Shape B.

*SRL-Left-1*



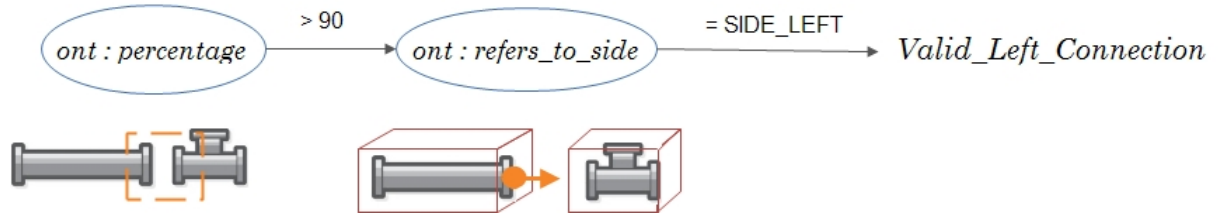


Fig. 5.2.14: Ejemplo de Spatial Relationship Rule

$$\begin{aligned}
 & (?i \text{ ont : percentage } ?x) \wedge \\
 & \quad \text{greaterThan}(?x, 90) \wedge \\
 & (?i \text{ ont : refers\_to\_side ont : SIDE\_LEFT}) \wedge \\
 & (?i \text{ rdf : typeont : Side\_Adjacency}) \\
 & \rightarrow (?i \text{ ont : isValid ont : TRUE})
 \end{aligned}$$

- Una conexión tipo **A Right B** es válida si, y sólo si, los puntos de la derecha del Shape A están 90% cerca de los puntos de la izquierda del Shape B  
**SRL-Right-2**

$$\begin{aligned}
 & (?i \text{ ont : percentage } ?x) \wedge \\
 & \quad \text{greaterThan}(?x, 90) \wedge \\
 & (?i \text{ ont : refers\_to\_side ont : SIDE\_RIGHT}) \wedge \\
 & (?i \text{ rdf : typeont : Side\_Adjacency}) \\
 & \rightarrow (?i \text{ ont : isValid ont : TRUE})
 \end{aligned}$$

- Una conexión tipo **A Front B** es válida si, y sólo si, los puntos del frente del Shape A están 90% cerca de los puntos posteriores del Shape B  
**SRL-Front-3**

$$\begin{aligned}
 & (?i \text{ ont : percentage } ?x) \wedge \\
 & \quad \text{greaterThan}(?x, 90) \wedge \\
 & (?i \text{ ont : refers\_to\_side ont : SIDE\_FRONT}) \wedge \\
 & (?i \text{ rdf : typeont : Side\_Adjacency}) \\
 & \rightarrow (?i \text{ ont : isValid ont : TRUE})
 \end{aligned}$$

- Una conexión tipo **A Back B** es válida si, y sólo si, los puntos traseros del Shape A están 90% cerca de los puntos frontales del Shape B  
**SRL-Back-4**

$$\begin{aligned}
 & (?i \text{ ont} : \text{percentage } ?x) \wedge \\
 & \quad \text{greaterThan}(?x, 90) \wedge \\
 & (?i \text{ ont} : \text{refers\_to\_side ont} : \text{SIDE\_BACK}) \wedge \\
 & (?i \text{ rdf} : \text{typeont} : \text{Side\_Adjacency}) \\
 & \rightarrow (?i \text{ ont} : \text{isValid ont} : \text{TRUE})
 \end{aligned}$$

- Una conexión tipo **A Above B** es válida si, y sólo si, los puntos por encima del Shape A están 90% cerca de los puntos bajos del Shape B  
**SRL-Above-5**

$$\begin{aligned}
 & (?i \text{ ont} : \text{percentage } ?x) \wedge \\
 & \quad \text{greaterThan}(?x, 90) \wedge \\
 & (?i \text{ ont} : \text{refers\_to\_side ont} : \text{SIDE\_ABOVE}) \wedge \\
 & (?i \text{ rdf} : \text{typeont} : \text{Side\_Adjacency}) \\
 & \rightarrow (?i \text{ ont} : \text{isValid ont} : \text{TRUE})
 \end{aligned}$$

- Una conexión tipo **A Below B** es válida si, y sólo si, los puntos por debajo del Shape A están 90% cerca de los puntos superiores del Shape B  
**SRL-Below-6**

$$\begin{aligned}
 & (?i \text{ ont} : \text{percentage } ?x) \wedge \\
 & \quad \text{greaterThan}(?x, 90) \wedge \\
 & (?i \text{ ont} : \text{refers\_to\_side ont} : \text{SIDE\_BELOW}) \wedge \\
 & (?i \text{ rdf} : \text{typeont} : \text{Side\_Adjacency}) \\
 & \rightarrow (?i \text{ ont} : \text{isValid ont} : \text{TRUE})
 \end{aligned}$$

### 5.2.2.6 Cálculo de la topología: representaciones espaciales

En el caso de las escenas industriales, las relaciones espaciales que se pueden suscitar en la vida real son Meet, Overlap, Equal, Inside and Contains, por lo que estas cinco relaciones espaciales fueron consideradas para esta tesis. La Tabla 5.2.15 muestra las relaciones topológicas que pueden existir en escenas industriales.

Tabla 5.2.15: Relaciones topológicas posibles de una escena industrial

Relaciones espaciales	<i>Meet</i>	<i>Overlap</i>	<i>Equal</i>	<i>Inside</i>	<i>Contains</i>
Pipe	Elbow Tee	Elbow Tee Pipe	Pipe	Pipe	Pipe
Elbow	Pipe	Pipe	Elbow	Elbow	Elbow
Tee	Pipe Valve	Pipe Valve	Tee	Tee	Tee
Valve	Tee	Tee	Valve	Valve	Valve

La manera de solventar estas condicionales es por medio de la implementación de reglas semánticas denominadas *Spatial Representation Rules*, las cuales son presentadas a continuación. Al igual que las *Spatial Relationship Rules*, cada una se describe brevemente y se agrega la regla en sintaxis de Jena. La Figura 5.2.15 representa gráficamente un ejemplo de este tipo de reglas semánticas.

- Una representación espacial **Meet** y **Overlap** de un **Pipe** es válida si, y sólo si, el tubo tiene conexiones libres, si la Relatum es un codo, un tubo o una te y si existe una relación global válida entre ellos del 10% de sus puntos.

#### *SRP-Pipe-Elbow-1*

```
(?i rdf : type ont : SpatialRelation) ^
(?i ont : has_a_relatum ?rel) ^
(?i ont : has_a_referent ?ref) ^
(?rel rdf : type ont : Elbow) ^
(?ref rdf : type ont : Pipe) ^
(?ref ont : number_connections ?num) ^
lessThan(?num, 3) ^
(?i ont : has_some ?adj) ^
(?adj rdf : type ont : Total_Adjacency) ^
(?adj ont : percentage ?per) ^
```

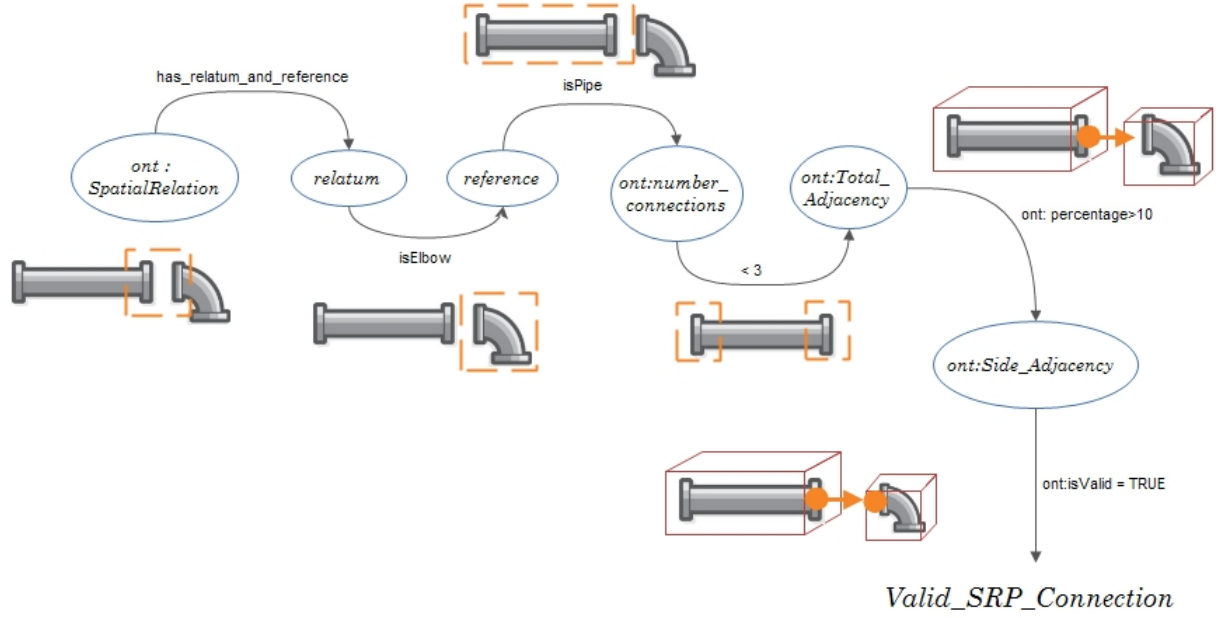


Fig. 5.2.15: Ejemplo de Spatial Representation Rule

$$\begin{aligned}
 & greaterThan(?per, 10) \wedge \\
 & (?i \text{ ont : has\_some } ?ads) \wedge \\
 & (?ads \text{ rdf : type ont : Side\_Adjacency}) \wedge \\
 & (?ads \text{ ont : isValid ont : TRUE}) \wedge \\
 & \rightarrow (?adj \text{ ont : isValid ont : TRUE})
 \end{aligned}$$

### *SRP-Pipe-Tee-2*

$$\begin{aligned}
 & (?i \text{ rdf : type ont : SpatialRelation}) \wedge \\
 & (?i \text{ ont : has\_a\_relatum } ?rel) \wedge \\
 & (?i \text{ ont : has\_a\_referent } ?ref) \wedge \\
 & (?rel \text{ rdf : type ont : Tee}) \wedge \\
 & (?ref \text{ rdf : type ont : Pipe}) \wedge \\
 & (?ref \text{ ont : number\_connections } ?num) \wedge \\
 & lessThan(?num, 3) \wedge \\
 & (?i \text{ ont : has\_some } ?adj) \wedge \\
 & (?adj \text{ rdf : type ont : Total\_Adjacency}) \wedge
 \end{aligned}$$



$$\begin{aligned}
& (?adj \text{ ont : percentage } ?per) ^\wedge \\
& \quad greaterThan(?per, 10) ^\wedge \\
& \quad (?i \text{ ont : has\_some } ?adjs) ^\wedge \\
& (?adjs \text{ rdf : type ont : Side\_Adjacency}) ^\wedge \\
& \quad (?adjs \text{ ont : isValid ont : TRUE}) ^\wedge \\
& \rightarrow (?adj \text{ ont : isValid ont : TRUE})
\end{aligned}$$

### ***SRP-Pipe-Pipe-3***

$$\begin{aligned}
& (?i \text{ rdf : type ont : SpatialRelation}) ^\wedge \\
& \quad (?i \text{ ont : has\_a\_relatum } ?rel) ^\wedge \\
& \quad (?i \text{ ont : has\_a\_referent } ?ref) ^\wedge \\
& \quad \quad (?rel \text{ rdf : type ont : Pipe}) ^\wedge \\
& \quad \quad (?ref \text{ rdf : type ont : Pipe}) ^\wedge \\
& (?ref \text{ ont : number\_connections } ?num) ^\wedge \\
& \quad lessThan(?num, 3) ^\wedge \\
& \quad (?i \text{ ont : has\_some } ?adj) ^\wedge \\
& (?adj \text{ rdf : type ont : Total\_Adjacency}) ^\wedge \\
& \quad (?adj \text{ ont : percentage } ?per) ^\wedge \\
& \quad \quad greaterThan(?per, 10) ^\wedge \\
& \quad \quad (?i \text{ ont : has\_some } ?adjs) ^\wedge \\
& (?adjs \text{ rdf : type ont : Side\_Adjacency}) ^\wedge \\
& \quad (?adjs \text{ ont : isValid ont : TRUE}) ^\wedge \\
& \rightarrow (?adj \text{ ont : isValid ont : TRUE})
\end{aligned}$$

- Una representación espacial **Meet y Overlap de un Elbow** es válida si y sólo si el codo tiene conexiones libres, si el Relatum es un Pipe, un Elbow o un Tee y si existe una relación global válida entre ellos el 10% de sus puntos.

### ***SRP-Elbow-Pipe-4***

$$\begin{aligned}
& (?i \text{ rdf : type ont : SpatialRelation}) ^\wedge \\
& \quad (?i \text{ ont : has\_a\_relatum } ?rel) ^\wedge \\
& \quad (?i \text{ ont : has\_a\_referent } ?ref) ^\wedge
\end{aligned}$$

$$\begin{aligned}
& (?rel \text{ rdf : type ont : Pipe} ) ^\wedge \\
& (?ref \text{ rdf : type ont : Elbow} ) ^\wedge \\
& (?ref \text{ ont : number\_connections ?num} ) ^\wedge \\
& \quad lessThan(?num, 3) ^\wedge \\
& (?i \text{ ont : has\_some ?adj} ) ^\wedge \\
& (?adj \text{ rdf : type ont : Total\_Adjacency} ) ^\wedge \\
& (?adj \text{ ont : percentage ?per} ) ^\wedge \\
& \quad greaterThan(?per, 10) ^\wedge \\
& (?i \text{ ont : has\_some ?adjs} ) ^\wedge \\
& (?adjs \text{ rdf : type ont : Side\_Adjacency} ) ^\wedge \\
& (?adjs \text{ ont : isValid ont : TRUE} ) ^\wedge \\
& \rightarrow (?adj \text{ ont : isValid ont : TRUE} )
\end{aligned}$$
***SRP-Elbow-Tee-5***

$$\begin{aligned}
& (?i \text{ rdf : type ont : SpatialRelation} ) ^\wedge \\
& (?i \text{ ont : has\_a\_relatum ?rel} ) ^\wedge \\
& (?i \text{ ont : has\_a\_referent ?ref} ) ^\wedge \\
& (?rel \text{ rdf : type ont : Tee} ) ^\wedge \\
& (?ref \text{ rdf : type ont : Elbow} ) ^\wedge \\
& (?ref \text{ ont : number\_connections ?num} ) ^\wedge \\
& \quad lessThan(?num, 3) ^\wedge \\
& (?i \text{ ont : has\_some ?adj} ) ^\wedge \\
& (?adj \text{ rdf : type ont : Total\_Adjacency} ) ^\wedge \\
& (?adj \text{ ont : percentage ?per} ) ^\wedge \\
& \quad greaterThan(?per, 10) ^\wedge \\
& (?i \text{ ont : has\_some ?adjs} ) ^\wedge \\
& (?adjs \text{ rdf : type ont : Side\_Adjacency} ) ^\wedge \\
& (?adjs \text{ ont : isValid ont : TRUE} ) ^\wedge \\
& \rightarrow (?adj \text{ ont : isValid ont : TRUE} )
\end{aligned}$$
***SRP-Elbow-Elbow-6***

$$(?i \text{ rdf : type ont : SpatialRelation} ) ^\wedge$$

$$\begin{aligned}
& (?i \text{ ont} : \text{has\_a\_relatum } ?rel) ^\wedge \\
& (?i \text{ ont} : \text{has\_a\_referent } ?ref) ^\wedge \\
& (?rel \text{ rdf} : \text{type ont} : \text{Elbow}) ^\wedge \\
& (?ref \text{ rdf} : \text{type ont} : \text{Elbow}) ^\wedge \\
& (?ref \text{ ont} : \text{number\_connections } ?num) ^\wedge \\
& \quad \text{lessThan}(?num, 3) ^\wedge \\
& (?i \text{ ont} : \text{has\_some } ?adj) ^\wedge \\
& (?adj \text{ rdf} : \text{type ont} : \text{Total\_Adjacency}) ^\wedge \\
& (?adj \text{ ont} : \text{percentage } ?per) ^\wedge \\
& \quad \text{greaterThan}(?per, 10) ^\wedge \\
& (?i \text{ ont} : \text{has\_some } ?adjs) ^\wedge \\
& (?adjs \text{ rdf} : \text{type ont} : \text{Side\_Adjacency}) ^\wedge \\
& (?adjs \text{ ont} : \text{isValid ont} : \text{TRUE}) ^\wedge \\
& \rightarrow (?adj \text{ ont} : \text{isValid ont} : \text{TRUE})
\end{aligned}$$

- Una representación espacial **Meet y Overlap de un Tee** es válida si, y sólo si, la te tiene conexiones libres, si el Relatum es una combinación de
  - Dos Tubos y una válvula ó
  - Dos codos y una válvula ó
  - Un tubo, un codo y una válvula ó
  - Tres codos ó
  - Tres tubos

Y si hay una relación global válida entre ellos y 10% de sus puntos.

***SRP-Tee-Pipe-7***

$$\begin{aligned}
& (?i \text{ rdf} : \text{type ont} : \text{SpatialRelation}) ^\wedge \\
& (?i \text{ ont} : \text{has\_a\_relatum } ?rel) ^\wedge \\
& (?i \text{ ont} : \text{has\_a\_referent } ?ref) ^\wedge \\
& (?rel \text{ rdf} : \text{type ont} : \text{Pipe}) ^\wedge \\
& (?ref \text{ rdf} : \text{type ont} : \text{Tee}) ^\wedge \\
& (?ref \text{ ont} : \text{number\_connections } ?num) ^\wedge \\
& \quad \text{lessThan}(?num, 4) ^\wedge \\
& (?i \text{ ont} : \text{has\_some } ?adj) ^\wedge
\end{aligned}$$

$$\begin{aligned}
& (?adj \text{ rdf : type ont : Total\_Adjacency} ) ^\wedge \\
& \quad (?adj \text{ ont : percentage ?per} ) ^\wedge \\
& \quad \quad greaterThan(?per, 10) ^\wedge \\
& \quad \quad (?i \text{ ont : has\_some ?ads} ) ^\wedge \\
& (?ads \text{ rdf : type ont : Side\_Adjacency} ) ^\wedge \\
& \quad (?ads \text{ ont : isValid ont : TRUE} ) ^\wedge \\
& \rightarrow (?adj \text{ ont : isValid ont : TRUE} )
\end{aligned}$$
***SRP-Tee-Valve-8***

$$\begin{aligned}
& (?i \text{ rdf : type ont : SpatialRelation} ) ^\wedge \\
& \quad (?i \text{ ont : has\_a\_relatum ?rel} ) ^\wedge \\
& \quad (?i \text{ ont : has\_a\_referent ?ref} ) ^\wedge \\
& \quad \quad (?rel \text{ rdf : type ont : Valve} ) ^\wedge \\
& \quad \quad (?ref \text{ rdf : type ont : Tee} ) ^\wedge \\
& (?ref \text{ ont : number\_connections ?num} ) ^\wedge \\
& \quad \quad lessThan(?num, 4) ^\wedge \\
& \quad \quad (?i \text{ ont : has\_some ?adj} ) ^\wedge \\
& (?adj \text{ rdf : type ont : Total\_Adjacency} ) ^\wedge \\
& \quad (?adj \text{ ont : percentage ?per} ) ^\wedge \\
& \quad \quad greaterThan(?per, 10) ^\wedge \\
& \quad \quad (?i \text{ ont : has\_some ?ads} ) ^\wedge \\
& (?ads \text{ rdf : type ont : Side\_Adjacency} ) ^\wedge \\
& \quad (?ads \text{ ont : isValid ont : TRUE} ) ^\wedge \\
& \rightarrow (?adj \text{ ont : isValid ont : TRUE} )
\end{aligned}$$
***SRP-Tee-Elbow-9***

$$\begin{aligned}
& (?i \text{ rdf : type ont : SpatialRelation} ) ^\wedge \\
& \quad (?i \text{ ont : has\_a\_relatum ?rel} ) ^\wedge \\
& \quad (?i \text{ ont : has\_a\_referent ?ref} ) ^\wedge \\
& \quad \quad (?rel \text{ rdf : type ont : Elbow} ) ^\wedge \\
& \quad \quad (?ref \text{ rdf : type ont : Tee} ) ^\wedge
\end{aligned}$$

$$\begin{aligned}
& (?ref \text{ ont} : \text{number\_connections } ?num) \wedge \\
& \quad \text{lessThan}(?num, 4) \wedge \\
& \quad (?i \text{ ont} : \text{has\_some } ?adj) \wedge \\
& (?adj \text{ rdf} : \text{type ont} : \text{Total\_Adjacency}) \wedge \\
& \quad (?adj \text{ ont} : \text{percentage } ?per) \wedge \\
& \quad \text{greaterThan}(?per, 10) \wedge \\
& \quad (?i \text{ ont} : \text{has\_some } ?adjs) \wedge \\
& (?adjs \text{ rdf} : \text{type ont} : \text{Side\_Adjacency}) \wedge \\
& \quad (?adjs \text{ ont} : \text{isValid ont} : \text{TRUE}) \wedge \\
& \rightarrow (?adj \text{ ont} : \text{isValid ont} : \text{TRUE})
\end{aligned}$$

- Una representación espacial **Meet y Overlap de un Valve** es válida si, y sólo si, la válvula tiene conexiones libres, si el Relatum es un te y si existe una relación espacial Meet entre ellos y el 10% de sus puntos.

***SRP- Valve-Tee-10***

$$\begin{aligned}
& (?i \text{ rdf} : \text{type ont} : \text{SpatialRelation}) \wedge \\
& \quad (?i \text{ ont} : \text{has\_a\_relatum } ?rel) \wedge \\
& \quad (?i \text{ ont} : \text{has\_a\_referent } ?ref) \wedge \\
& \quad (?rel \text{ rdf} : \text{type ont} : \text{Tee}) \wedge \\
& \quad (?ref \text{ rdf} : \text{type ont} : \text{Valve}) \wedge \\
& (?ref \text{ ont} : \text{number\_connections } ?num) \wedge \\
& \quad \text{lessThan}(?num, 2) \wedge \\
& \quad (?i \text{ ont} : \text{has\_some } ?adj) \wedge \\
& (?adj \text{ rdf} : \text{type ont} : \text{Total\_Adjacency}) \wedge \\
& \quad (?adj \text{ ont} : \text{percentage } ?per) \wedge \\
& \quad \text{greaterThan}(?per, 10) \wedge \\
& \quad (?i \text{ ont} : \text{has\_some } ?adjs) \wedge \\
& (?adjs \text{ rdf} : \text{type ont} : \text{Side\_Adjacency}) \wedge \\
& \quad (?adjs \text{ ont} : \text{isValid ont} : \text{TRUE}) \wedge \\
& \rightarrow (?adj \text{ ont} : \text{isValid ont} : \text{TRUE})
\end{aligned}$$

### 5.2.2.7 Resultados del cálculo de topología

El resultado de cada una se describe a continuación. Para facilitar la comprensión del lector de las reglas aplicadas dentro del algoritmo a la ontología, se explicará en detalle el

comportamiento del segmento llamado *shape\_003*, extendiendo su actuación al resto de elementos pertenecientes a la nube de puntos. Para ello, se presenta que el segmento *shape\_003* fue clasificado por el Proceso de Reconocimiento de Primitivas como un cilindro debido a sus propiedades geométricas, a continuación, se aplicaron dos afirmaciones (*assert-1* y *assert-5*), las cuales afirman que este segmento es un tubo y tiene 2 conexiones permitidas.

Las reglas de Relaciones Espaciales (*Spatial Relationship Rules*) y de Representaciones Espaciales (*Spatial Representation Rules*) nos permiten definir la topología de cada segmento. Las afirmaciones, *Spatial Relationship Rules* y *Spatial Representation Rules* que se aplicaron a cada segmento de la nube de puntos de prueba se muestran en la Tabla 5.2.16.

Las *Spatial Relationship Rules* (SRL-Front-3 y SRL-Back-4) se aplicaron a Pipe representado por el *Shape\_003*, lo que demuestra que el tubo tiene una conexión válida de tipo A Front B con el tubo representado por *Shape\_023* y también una válida conexión de tipo A Back B con el Pipe en el *Shape\_004*. Mientras, las *Spatial Representation Rules* (SRP-Pipe-Pipe-3) muestran que el Pipe en el *Shape\_003* tiene un tipo de representación espacial Overlap con el tubo en el *Shape\_004*, debido a que se cumplieron las siguientes condiciones:

1. El relatum (*Shape\_023*) es un *Shape* del tipo *Pipe* y
2. Hay un tipo de relación espacial *Meet* entre ambos segmentos de más del 10% de sus puntos.

Y por último, un tipo de representación espacial *Meet* fue creado para el mismo segmento (*Shape\_003*), debido que se satisfacen las siguientes condiciones:

1. El relatum (*Shape\_004*) es un *Shape* del tipo *Pipe* y
2. Existe un tipo de relación espacial *Meet* entre el 10% de los puntos de ambas formas.

Tabla 5.2.16: Segmentos con afirmaciones y relaciones espaciales aplicadas

:Shape	Geometric Primitive :Shape	Factory's Element :Shape	:Shape. Points	Afirmaciones	Spatial Relation Rules	Spatial Representation Rules
shape_001	Cylinder	Pipe	57.472	Assert-1, Assert-5		
shape_002	Cylinder	Pipe	30.619	Assert-1, Assert-5		
shape_003	Cylinder	Pipe	18.705	Assert-1, Assert-5	SRL-Front-3, SRL-Back-4	SRP-Pipe-Pipe-3
Continúa en la siguiente página						



Tabla 5.2.16 – continuación de página previa

<i>:Shape</i>	<i>Geometric Primitive :Shape</i>	<i>Factory's Element :Shape</i>	<i>:Shape.Points</i>	<i>Afirmaciones</i>	<i>Spatial Relation Rules</i>	<i>Spatial Representation Rules</i>
shape_004	Cylinder	Pipe	15.158	Assert-1, Assert-5	SRL-Front-3	SRP-Pipe-Pipe-3
shape_005	Cylinder	Pipe	10623	Assert-1, Assert-5	SRL-Front-3, SRL-Back-4	SRP-Pipe-Pipe-3, SRP-Pipe-Elbow-1
shape_006	Cylinder	Pipe	10.019	Assert-1, Assert-5		
shape_007	Cylinder	Pipe	10715	Assert-1, Assert-5		
shape_008	Cylinder	Pipe	10.135	Assert-1, Assert-5	SRL-Above-5, SRL-Bellow-6	SRP-Pipe-Elbow-1, SRP-Pipe-Pipe-3
shape_009	Cylinder	Pipe	8519	Assert-1, Assert-5		
shape_010	Cylinder	Pipe	5812	Assert-1, Assert-5	SRL-Above-5	SRP-Pipe-Elbow-1, SRP-Pipe-Pipe-3
shape_011	Sphere	Tee	6560	Assert-4, Assert-8	SRL-Above-5, SRL-Bellow-6	SRP-Tee-Pipe-7, SRP-Tee-Elbow-9, SRP-21
shape_012	Cylinder	Pipe	5014	Assert-1, Assert-5	SRL-Front-3, SRL-Back-4	SRP-Pipe-Elbow-1, SRP-Pipe-Pipe-3
shape_013	Torus	Elbow	5454	Assert-2, Assert-6	SRL-Left-1, SRL-Bellow-6	SRP-Elbow-Elbow-6 SRP-Elbow-Tee-5
shape_014	Torus	Elbow	4709	Assert-2, Assert-6	SRL-Left-1, SRL-Right-2	SRP-Elbow-Tee-5, SRP-Elbow-Elbow-6
shape_015	Cylinder	Pipe	4313	Assert-1, Assert-5	SRL-Above-5, SRL-Bellow-6	SRP-Pipe-Pipe-3, SRP-21
shape_016	Cylinder	Pipe	3979	Assert-1, Assert-5	SRL-Front-3	SRP-Pipe-Tee-2
shape_017	Torus	Valve	4609	Assert-3, Assert-7	SRL-Bellow-6	SRP-Valve-Tee-10
shape_018	Sphere	Tee	3580	Assert-4, Assert-8	SRL-Bellow-6, SRL-Above-5	SRP-Pipe-Tee-2, SRP-21, SRP-Tee-Elbow-9
shape_019	Torus	Elbow	3292	Assert-2, Assert-6	SRL-Left-1, SRL-Bellow-6	SRP-Elbow-Tee-5, SRP-Elbow-Elbow-6
shape_020	Torus	Elbow	3140	Assert-2, Assert-6	SRL-Front-3, SRL-Bellow-6	SRP-Elbow-Pipe-4, SRP-Elbow-Elbow-6
Continúa en la siguiente página						

Tabla 5.2.16 – continuación de página previa

<i>:Shape</i>	<i>Geometric Primitive :Shape</i>	<i>Factory's Element :Shape</i>	<i>:Shape.Points</i>	<i>Afirmaciones</i>	<i>Spatial Relation Rules</i>	<i>Spatial Representation Rules</i>
shape_021	Torus	Elbow	3299	Assert-2, Assert-6	SRL-Bellow-6, SRL-Front-3	SRP- Elbow- Pipe-4
shape_022	Cylinder	Pipe	3028	Assert-1, Assert-5		
shape_023	Cylinder	Pipe	2709	Assert-1, Assert-5	SRL-Back- 4	SRP-Pipe- Pipe-3
shape_024	Cylinder	Pipe	3757	Assert-1, Assert-5	SRL-Back- 4	SRP-Pipe- Pipe-3
shape_025	Cylinder	Pipe	3227	Assert-1, Assert-5	SRL-Front-3, SRL-Back-4	SRP-Pipe-Pipe-3, SRP-Pipe-Elbow-1
shape_026	Sphere	Tee	2840	Assert-3, Assert-7		
shape_027	Cylinder	Pipe	2066	Assert-1, Assert-5		
shape_028	Sphere	Tee	2879	Assert-4, Assert-8	SRL-Front-3, SRL-Back-4, SRL-Above-5	SRP-Tee-Pipe-7, SRP-Tee-Valve-8, SRP-Tee-Elbow-9
shape_029	Cylinder	Pipe	2002	Assert-1, Assert-5		
shape_030	Cylinder	Pipe	2981	Assert-1, Assert-5	SRL-Back- 4	SRP-Pipe- Pipe-3
shape_032	Cylinder	Pipe	1737	Assert-1, Assert-5	SRL-Front-3,	SRP-Pipe- Pipe-3
shape_033	Cylinder	Pipe	1620	Assert-1, Assert-5	SRL-Left-1, SRL-Right-2	SRP-Pipe- Pipe-3
shape_034	Torus	Elbow	1598	Assert-2, Assert-6	SRL-Right-2, SRL-Bellow-6	SRP-Elbow-Pipe-4, SRP-Elbow-Elbow-6
shape_035	Torus	Elbow	1696	Assert-2, Assert-6	SRL-Right- 2	SRP- Elbow- Elbow-6
shape_036	Cylinder	Pipe	1777	Assert-1, Assert-5		
shape_037	Torus	Elbow	2448	Assert-2, Assert-6	SRL-Right-2, SRL-Back-4	SRP-Elbow-Elbow-6, SRP-Elbow-Elbow-6
Continúa en la siguiente página						



Tabla 5.2.16 – continuación de página previa

<i>:Shape</i>	<i>Geometric Primitive :Shape</i>	<i>Factory's Element :Shape</i>	<i>:Shape.Points</i>	<i>Afirmaciones</i>	<i>Spatial Relation Rules</i>	<i>Spatial Representation Rules</i>
shape_038	Cylinder	Pipe	1846	Assert-4, Assert-8	SRL-Front-3, SRL-Back-4	SRP-Pipe-Tee-2, SRP-Pipe-Elbow-1
shape_039	Cylinder	Pipe	1834	Assert-1, Assert-5	SRL-Left-1	
shape_040	Cylinder	Pipe	1400	Assert-1, Assert-5		
shape_041	Cylinder	Pipe	1127	Assert-1, Assert-5	SRL- Bellow-6	SRP-Pipe- Pipe-3
shape_042	Cylinder	Pipe	1863	Assert-1, Assert-5	SRL-Front- 3	SRP-Pipe- Pipe-3
shape_043	Torus	Elbow	1027	Assert-2, Assert-6	SRL-Front- 3	SRP- Elbow- Elbow-6
shape_044	Cylinder	Pipe	956	Assert-1, Assert-5	SRL-Above-5, SRL-Bellow-6	SRP-Pipe- Pipe-3
shape_045	Cylinder	Pipe	156	Assert-1, Assert-5		
shape_046	Cylinder	Pipe	1291	Assert-2, Assert-6	SRL-Left-1	SRP-Pipe- Pipe-3
shape_047	Sphere	Tee	824	Assert-1, Assert-5	SRL-Bellow-6, SRL-Above-5	SRP-Tee- Pipe-7
shape_048	Torus	Elbow	856	Assert-2, Assert-6		
shape_049	Torus	Elbow	1042	Assert-2, Assert-6	SRL-Front-3, SRL-Back-4, SRL-Bellow-6	SRP- Elbow- Pipe-4
shape_050	Torus	Valve	1222	Assert-4, Assert-8		SRP-21
shape_051	Cylinder	Pipe	753	Assert-1, Assert-5	SRL-Back- 4	SRP-Pipe- Pipe-3
shape_052	Sphere	Tee	137	Assert-4, Assert-8	SRL-Bellow-6, SRL-Above-5	SRP-Tee- Pipe-7
shape_053	Cylinder	Pipe	779	Assert-1, Assert-5	SRL-Front- 3	SRP-Pipe- Pipe-3
Continúa en la siguiente página						

Tabla 5.2.16 – continuación de página previa

<i>:Shape</i>	<i>Geometric Primitive :Shape</i>	<i>Factory's Element :Shape</i>	<i>:Shape.Points</i>	<i>Afirmaciones</i>	<i>Spatial Relation Rules</i>	<i>Spatial Representation Rules</i>
shape_054	Torus	Elbow	674	Assert-1, Assert-5	SRL-Left-1, SRL-Right-2	SRP- Elbow- Elbow-6
shape_055	Torus	Valve	718	Assert-4, Assert-8	SRL- Bellow-6	SRP-Valve- Tee-10
shape_056	Cylinder	Pipe	642	Assert-1, Assert-5		
shape_057	Cylinder	Pipe	923	Assert-1, Assert-5	SRL-Back- 4	SRP-Pipe- Pipe-3
shape_058	Torus	Elbow	594	Assert-1, Assert-5	SRL-Right-2, SRL-Front-3	SRP-Elbow-Pipe-4, SRP-Elbow-Elbow-6

Una mejor manera de mostrar las relaciones entre los segmentos es por medio de gráficos. La Figura 5.2.16 muestra un ejemplo de las relaciones asociadas al *Pipe* mediante la implementación de las afirmaciones y las *Spatial Relation Rules*. Donde el Shape\_003 tiene permitido solo por relaciones (Assert-5), estas relaciones están descritas por medio de una relación espacial *FRONT* con el *Shape* tipo *Pipe* en Shape\_023 y una relación espacial *BACK* con el *Shape* de tipo *Pipe* en Shape\_004.

De la misma forma, la Figura 5.2.17 muestra un ejemplo de las relaciones asociadas de un *Elbow* mediante la implementación las reglas y afirmaciones descritas anteriormente. El *Elbow* representado por el Shape\_034 tiene la restricción de solo 2 relaciones espaciales de acuerdo con la Assert-6. Por lo que las relaciones asociadas describen un relación espacial *BELOW* con un *Shape* tipo *Pipe* en Shape\_008 y una relación espacial *RIGHT* con el *Shape* tipo *Elbow* representado en Shape\_014.

El tercer ejemplo se muestra en la Figura 5.2.18, donde se presentan las relaciones asociadas con un *Tee* mediante la aplicación de las normas y afirmaciones presentadas en esta tesis. El *Tee* representado en el Shape\_028, tiene la restricción de solo 3 relaciones espaciales mediante la Assert-8. La primer relación espacial es *FRONT* con un *Shape* tipo *Elbow* presente en el Shape\_014, una relación espacial *BELOW* con un *Shape* tipo *Valve* en el Shape\_055, y finalmente una relación espacial *BACK* con un *Shape* tipo *Pipe* en Shape\_038.

#### 5.2.2.8 Validación semántica

Como ya se ha mencionado, este proceso verifica semánticamente la clasificación creada. Por medio de reglas semánticas llamadas *textitRedundant Shape Rules* se elimina el prob-

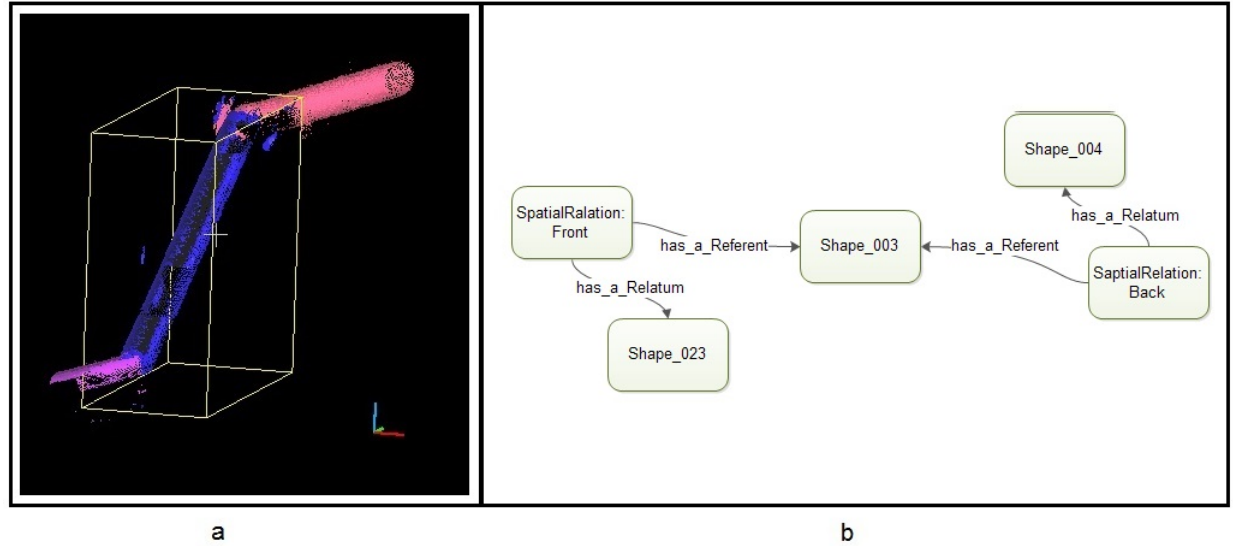


Fig. 5.2.16: Dos relaciones espaciales de un segmento tipo *Pipe* en Shape\_003, donde existe una relación espacial FRONT con un *Pipe* en Shape\_023 y una relación espacial BACK con un *Pipe* en Shape\_004. **a)** Nube de puntos de los tres segmentos, **b)** grafo de las relaciones

lema de sobre-segmentación por solapamiento. La Figura 5.2.19 presenta un ejemplo de este problema y de la regla a aplicar se presenta en en mismo.



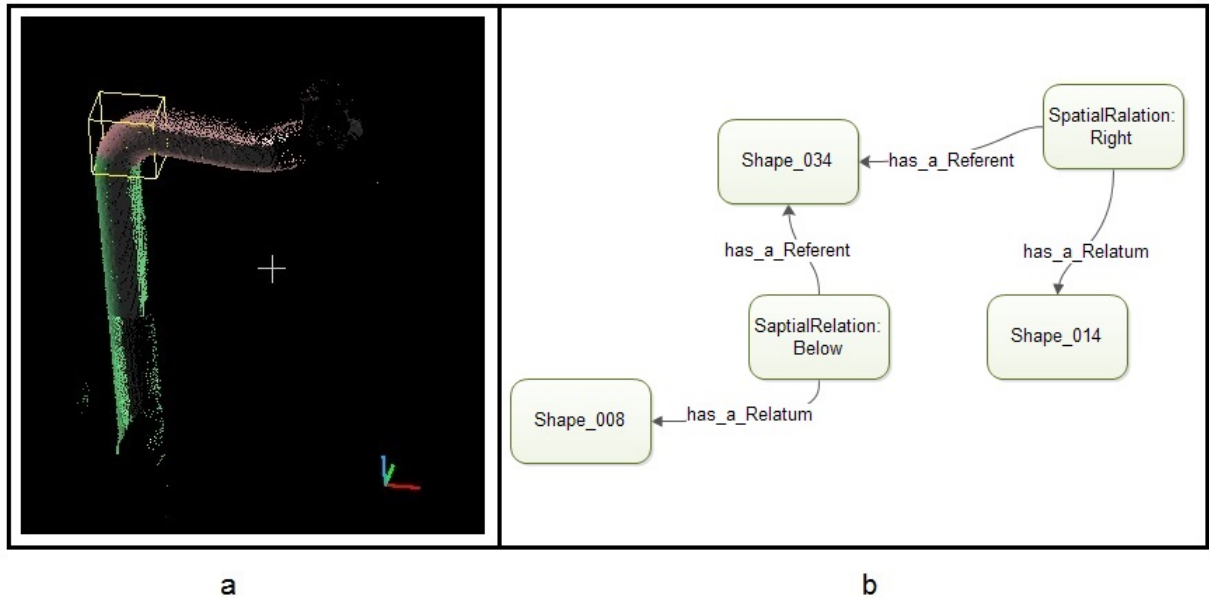


Fig. 5.2.17: Dos relaciones espaciales de un segmento tipo *Elbow* en Shape\_34, el cual tiene una relación espacial BELOW con un Pipe en Shape\_008 y una relación espacial RIGHT con un *Elbow* en Shape\_014. **a)** Nube de puntos de los tres segmentos, **b)** grafo de las relaciones

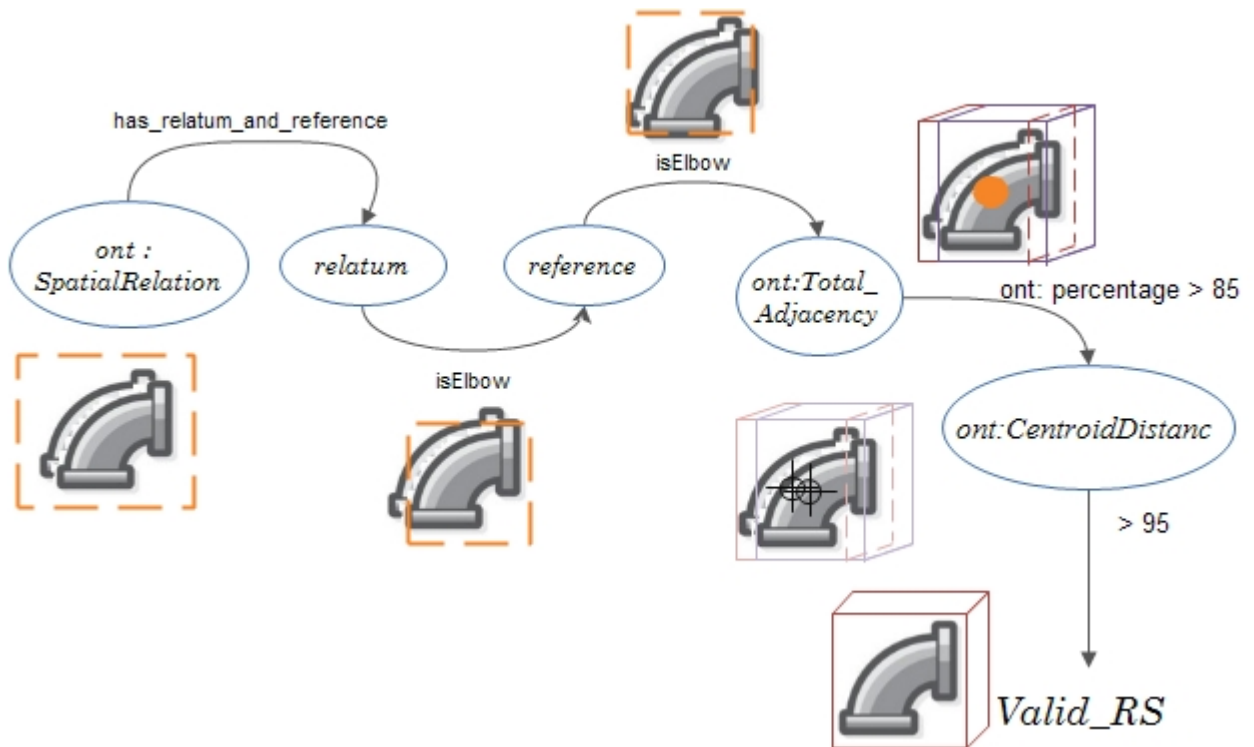


Fig. 5.2.19: Ejemplo de Redundant Shape Rules

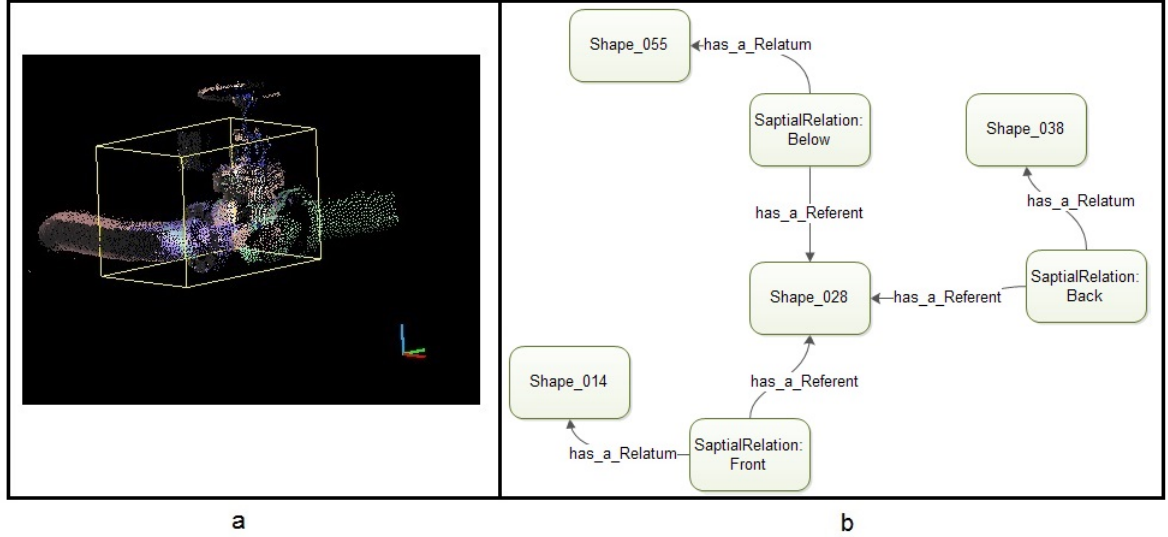


Fig. 5.2.18: Tres relaciones espaciales del segmento tipo *Tee* en el Shape\_028, el cual tiene un restricción de solo 3 relaciones espaciales: FRONT con un *Elbow* en el Shape\_01, una relación espacial BACK con el *Shape* tipo *Pipe* en el Shape\_004 y una relación espacial BELOW con un *Shape* tipo *Valve* en Shape\_055. **a)** Nube de puntos de los tres segmentos, **b)** grafo de las relaciones

La lista de *Redundant Shape Rules* se muestran a continuación.

- Combinar segmentos A y B si, y sólo si, la Refent A y el Relatum B son del mismo tipo, sus centroides se encuentran 95% cercanos, y todos los puntos están 85% cercanos.

### ***SRP-Pipe-Pipe-21***

```
(?i rdf : type ont : SpatialRelation) ^
  (?i ont : has_a_relatum ?rel) ^
  (?i ont : has_a_referent ?ref) ^
  (?rel rdf : type ont : Pipe) ^
  (?ref rdf : type ont : Pipe) ^
  (?i ont : has_some ?adj) ^
  (?adj rdf : type ont : TotalAdjacency) ^
```

$$\begin{aligned}
& (?adj \text{ ont : percentage } ?per) \wedge \\
& \quad greaterThan(?per, 85) \wedge \\
& (?rel \text{ ont : has\_some } ?cd) \wedge \\
& (?cd \text{ rdf : type ont : CentroidDistance}) \wedge \\
& (?cd \text{ ont : percentage } ?cdPerc) \wedge \\
& \quad greaterThan(?cdPerc, 95) \wedge \\
& \rightarrow (?adj \text{ ont : isValid ont : TRUE})
\end{aligned}$$
***SRP-Elbow-Elbow***

$$\begin{aligned}
& (?i \text{ rdf : type ont : SpatialRelation}) \wedge \\
& (?i \text{ ont : has\_a\_relatum } ?rel) \wedge \\
& (?i \text{ ont : has\_a\_referent } ?ref) \wedge \\
& (?rel \text{ rdf : type ont : Elbow}) \wedge \\
& (?ref \text{ rdf : type ont : Elbow}) \wedge \\
& (?i \text{ ont : has\_some } ?adj) \wedge \\
& (?adj \text{ rdf : type ont : TotalAdjacency}) \wedge \\
& (?adj \text{ ont : percentage } ?per) \wedge \\
& \quad greaterThan(?per, 85) \wedge \\
& (?rel \text{ ont : has\_some } ?cd) \wedge \\
& (?cd \text{ rdf : type ont : CentroidDistance}) \wedge \\
& (?cd \text{ ont : percentage } ?cdPerc) \wedge \\
& \quad greaterThan(?cdPerc, 95) \wedge \\
& \rightarrow (?adj \text{ ont : isValid ont : TRUE})
\end{aligned}$$
***SRP-Tee-Tee***

$$\begin{aligned}
& (?i \text{ rdf : type ont : SpatialRelation}) \wedge \\
& (?i \text{ ont : has\_a\_relatum } ?rel) \wedge \\
& (?i \text{ ont : has\_a\_referent } ?ref) \wedge \\
& (?rel \text{ rdf : type ont : Tee}) \wedge \\
& (?ref \text{ rdf : type ont : Tee}) \wedge \\
& (?i \text{ ont : has\_some } ?adj) \wedge
\end{aligned}$$

$$\begin{aligned}
& (?adj \text{ rdf : type ont : TotalAdjacency} ) ^\wedge \\
& \quad (?adj \text{ ont : percentage ?per} ) ^\wedge \\
& \quad \quad greaterThan(?per, 85) ^\wedge \\
& \quad \quad (?rel \text{ ont : has\_some ?cd} ) ^\wedge \\
& (?cd \text{ rdf : type ont : CentroidDistance} ) ^\wedge \\
& \quad (?cd \text{ ont : percentage ?cdPerc} ) ^\wedge \\
& \quad \quad greaterThan(?cdPerc, 95) ^\wedge \\
& \rightarrow (?adj \text{ ont : isValid ont : TRUE} )
\end{aligned}$$

### ***SRP-Valve-Valve***

$$\begin{aligned}
& (?i \text{ rdf : type ont : SpatialRelation} ) ^\wedge \\
& \quad (?i \text{ ont : has\_a\_relatum ?rel} ) ^\wedge \\
& \quad (?i \text{ ont : has\_a\_referent ?ref} ) ^\wedge \\
& \quad \quad (?rel \text{ rdf : type ont : Valve} ) ^\wedge \\
& \quad \quad (?ref \text{ rdf : type ont : Valve} ) ^\wedge \\
& \quad \quad \quad (?i \text{ ont : has\_some ?adj} ) ^\wedge \\
& (?adj \text{ rdf : type ont : TotalAdjacency} ) ^\wedge \\
& \quad (?adj \text{ ont : percentage ?per} ) ^\wedge \\
& \quad \quad greaterThan(?per, 85) ^\wedge \\
& \quad \quad (?rel \text{ ont : has\_some ?cd} ) ^\wedge \\
& (?cd \text{ rdf : type ont : CentroidDistance} ) ^\wedge \\
& \quad (?cd \text{ ont : percentage ?cdPerc} ) ^\wedge \\
& \quad \quad greaterThan(?cdPerc, 95) ^\wedge \\
& \rightarrow (?adj \text{ ont : isValid ont : TRUE} )
\end{aligned}$$

Por otra parte, la sobre-segmentación por unión es solventada a través de su localización y unión en un nuevo segmento. Esta condición se observó solo en los elementos cilíndricos por que que la regla solo aplica a dichos elementos. La Figura 5.2.20 muestra una descripción gráfica de la regla.

- Unir segmentos si, y sólo si, el Referent A y el Relatum B son tubos y tienen una representación espacial Meet u Overlap.

### ***SRP-Pipe-Pipe-24***

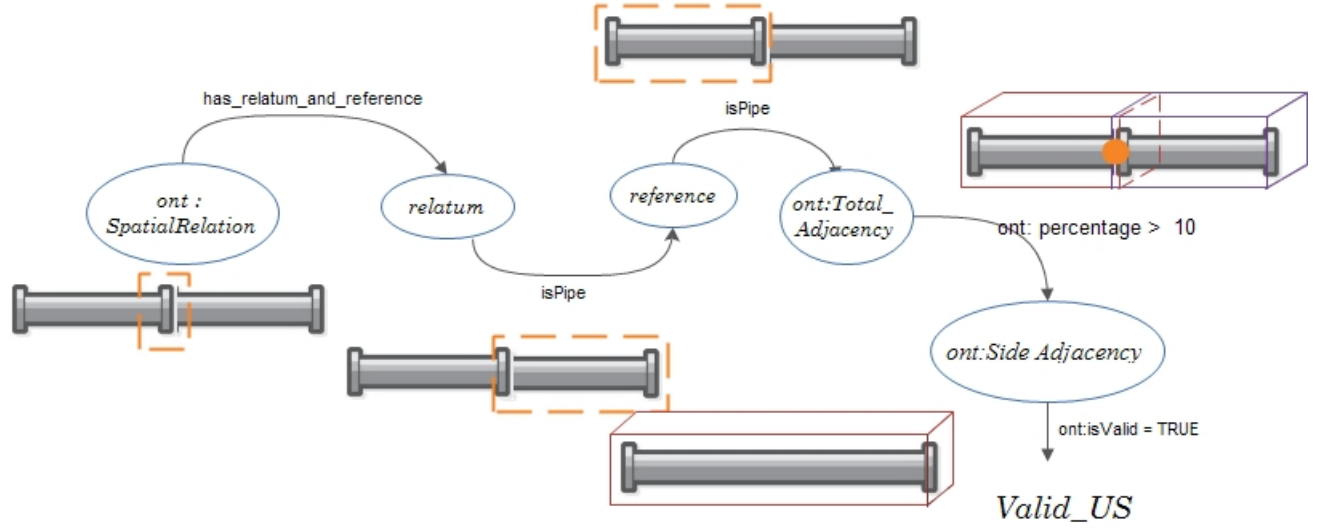


Fig. 5.2.20: Ejemplo de Union Rules

```

(?i rdf : type ont : SpatialRelation) ^
  (?i ont : has_a_relatum ?rel) ^
  (?i ont : has_a_referent ?ref) ^
  (?rel rdf : type ont : Pipe) ^
  (?ref rdf : type ont : Pipe) ^
  (?ref ont : number_connections ?num) ^
    lessThan(?num, 3) ^
  (?i ont : has_some ?adj) ^
  (?adj rdf : type ont : Total_Adjacency) ^
  (?adj ont : percentage ?per) ^
    greaterThan(?per, 10) ^
  (?i ont : has_some ?adjs) ^
  (?adjs rdf : type ont : Side_Adjacency) ^
  (?adjs ont : isValid ont : TRUE) ^
  → (?adj ont : isValid ont : TRUE)

```



### 5.2.2.9 Resultados del cálculo de validación semántica

Las *Redundant Shape Rules* y *Union Rules* permiten decidir si dos segmentos pueden unirse para formar un segmento común. Las relaciones creadas por la aplicación de estas normas a los segmentos se muestran en la Tabla 5.2.17.

La regla de Unión de Formas (SRP-24) se aplicó al segmento Shape\_003, cumpliendo con las siguientes condiciones:

1. El Refent A (Pipe en Shape\_003) y el relatum B (Pipe en Shape\_004) ambos pertenecen a la misma clase de formas, es decir *Pipes* y
2. Existe una representación espacial Meet entre ambas formas,

Por otra parte, la regla SRP-25 se aplicó también, debido a que las siguientes condiciones se satisfacen:

1. El Refent A (Pipe en Shape\_003) y el relatum B (Pipe en Shape\_023), ambas pertenecen a la mismo tipo, esto es, son *Pipes* y
2. existe un tipo de representación espacial Overlap entre ambas formas. Por lo tanto, los tres segmentos (Shape\_003, Shape\_004 y Shape\_023) se unieron para crear el segmento en Shape $\mathcal{U}1$ .

Tabla 5.2.17: Segmentos con *Redundant Shape Rules* y *Union Rules*

<i>:Spatial Relation. Referent</i>	<i>:Spatial Relation. Relatum</i>	<i>:SpatialRelation. GlobalRelationship</i>	<i>:SpatialRelation. SpatialRepresentation</i>	<i>Rules</i>
shape_003	shape_023	Front	Overlap	SRP-25
shape_003	shape_004	Back	Meet	SRP-24
shape_004	shape_003	Front	Meet	SRP-24
shape_005	shape_012	Front	Meet	SRP-24
shape_011	shape_018		Equals	SRP-21
shape_017	shape_055		Equals	SRP-21
shape_018	shape_026		Equals	SRP-21
shape_018	shape_011		Equals	SRP-21
shape_019	shape_054		Contains	SRP-23
shape_023	shape_003	Back	Overlap	SRP-25
shape_026	shape_011		Equals	SRP-21
shape_026	shape_018		Equals	SRP-21
shape_030	shape_042	Back	Overlap	SRP-25
shape_030	shape_041		Contains	SRP-23

Continúa en la siguiente página

Tabla 5.2.17 – continuación de página previa

<i>:Spatial Relation. Referent</i>	<i>:Spatial Relation. Relatum</i>	<i>:SpatialRelation. GlobalRelationship</i>	<i>:SpatialRelation. SpatialRepresentation</i>	<i>Rules</i>
shape_030	shape_036		Contains	SRP-23
shape_030	shape_039		Contains	SRP-23
shape_030	shape_045		Contains	SRP-23
shape_030	shape_046		Equals	SRP-21
shape_032	shape_051	Front	Meet	SRP-24
shape_036	shape_039		Equals	SRP-21
shape_036	shape_046		Inside	SRP-22
shape_036	shape_030		Inside	SRP-22
shape_039	shape_036		Equals	SRP-21
shape_039	shape_042	Left	Meet	SRP-24
shape_039	shape_045		Contains	SRP-23
shape_039	shape_046		Equals	SRP-21
shape_039	shape_030		Inside	SRP-22
shape_041	shape_030		Inside	SRP-22
shape_042	shape_030	Front	Overlap	SRP-25
shape_045	shape_039		Inside	SRP-22
shape_045	shape_046		Inside	SRP-22
shape_045	shape_057		Inside	SRP-22
shape_045	shape_030		Inside	SRP-22
shape_046	shape_030		Equals	SRP-21
shape_046	shape_039		Equals	SRP-21
shape_047	shape_052		Equals	SRP-21
shape_051	shape_032	Back	Meet	SRP-24
shape_052	shape_047		Equals	SRP-21
shape_053	shape_042		Inside	SRP-22
shape_053	shape_057	Front	Meet	SRP-24
shape_055	shape_017		Equals	SRP-21
shape_057	shape_053	Back	Meet	SRP-24

Finalmente, los segmentos resultantes después de haber aplicado los tres tipos de reglas a la nube de puntos de este caso de uso real se muestran en la Tabla 5.2.18, donde es posible ver los ocho nuevas uniones. A su vez, nuevas relaciones fueron creadas y actualizadas, las relaciones resultantes de los segmentos se muestran en la Tabla 5.2.19.



Tabla 5.2.18: Segmentos finales

<i>:Shape</i>	<i>Geometric Primitive :Shape</i>	<i>Factory's Element :Shape</i>	<i>:Shape Points</i>
shape_001	Cylinder	Pipe	57472
shape_002	Cylinder	Pipe	30619
shape_U1	Cylinder	Pipe	36572
shape_U2	Cylinder	Pipe	15637
shape_006	Cylinder	Pipe	10019
shape_007	Cylinder	Pipe	10715
shape_008	Cylinder	Pipe	10135
shape_009	Cylinder	Pipe	8519
shape_010	Cylinder	Pipe	5812
shape_U3	Sphere	Tee	12980
shape_013	Torus	Elbow	5454
shape_014	Torus	Elbow	4709
shape_015	Cylinder	Pipe	4313
shape_016	Cylinder	Pipe	3979
shape_U4	Sphere	Valve	5327
shape_U5	Torus	Elbow	3966
shape_020	Torus	Elbow	3140
shape_021	Torus	Elbow	3299
shape_022	Cylinder	Pipe	3028
shape_024	Cylinder	Pipe	3757
shape_025	Cylinder	Pipe	3227
shape_026	Torus	Tee	2840
shape_027	Cylinder	Pipe	2066
shape_028	Torus	Tee	2879
shape_029	Cylinder	Pipe	2002
shape_U6	Cylinder	Pipe	11029
shape_U7	Cylinder	Pipe	2490
shape_033	Cylinder	Pipe	1620
shape_034	Torus	Elbow	1598
shape_035	Cylinder	Elbow	1696
shape_037	Torus	Elbow	2448
shape_038	Cylinder	Pipe	1846
shape_040	Cylinder	Pipe	1400
shape_043	Torus	Elbow	1027
Continúa en la siguiente página			

Tabla 5.2.18 – continuación de página previa

<i>:Shape</i>	<i>Geometric Primitive :Shape</i>	<i>Factory's Element :Shape</i>	<i>:Shape Points</i>
shape_044	Cylinder	Pipe	956
shape_U8	Torus	Tee	961
shape_048	Torus	Elbow	856
shape_049	Torus	Elbow	1042
shape_050	Sphere	Valve	1222
shape_053	Cylinder	Pipe	779
shape_054	Torus	Elbow	674
shape_056	Cylinder	Pipe	642
shape_057	Cylinder	Pipe	923
shape_058	Torus	Elbow	594

Tabla 5.2.19: Segmentos finales y sus relaciones

<i>:SpatialRelation .Referent</i>	<i>:SpatialRelation .Relatum</i>	<i>:SpatialRelation. SpatialRepresentation</i>	<i>:SpatialRelation. GlobalRepresentation</i>
shape_U2	shape_021	Back	Meet
shape_008	shape_034	Above	Meet
shape_008	shape_016	Below	Meet
shape_010	shape_049	Above	Meet
shape_010	shape_024	Below	Meet
shape_U3	shape_U5	Above	Meet
shape_U3	shape_016	Below	Meet
shape_U2	shape_049	Front	Meet
shape_013	shape_058	Left	Overlap
shape_013	shape_002	Below	Meet
shape_014	shape_028	Back	Overlap
shape_014	shape_034	Left	Meet
shape_015	shape_033	Above	Meet
shape_015	shape_022	Below	Meet
shape_016	shape_U3	Back	Meet
shape_U4	shape_028	Below	Overlap
shape_U5	shape_037	Left	Overlap
shape_U5	shape_U3	Below	Overlap
shape_020	shape_037	Front	Meet

Continúa en la siguiente página

Tabla 5.2.19 – continuación de página previa

<i>:SpatialRelation</i> <i>.Referent</i>	<i>:SpatialRelation</i> <i>.Relatum</i>	<i>:SpatialRelation.</i> <i>SpatialRepresentation</i>	<i>:SpatialRelation.</i> <i>GlobalRepresentation</i>
shape_020	shape_009	Below	Meet
shape_021	shape_U2	Front	Meet
shape_021	shape_006	Below	Meet
shape_024	shape_010	Back	Meet
shape_025	shape_033	Front	Meet
shape_025	shape_054	Back	Meet
shape_028	shape_014	Front	Overlap
shape_028	shape_038	Back	Overlap
shape_028	shape_U4	Above	Overlap
shape_U7	shape_001	Back	Meet
shape_033	shape_025	Right	Meet
shape_033	shape_015	Left	Meet
shape_034	shape_008	Below	Meet
shape_034	shape_014	Right	Meet
shape_035	shape_043	Right	Meet
shape_037	shape_U5	Right	Overlap
shape_037	shape_020	Back	Meet
shape_038	shape_028	Front	Overlap
shape_038	shape_058	Back	Meet
shape_U6	shape_024	Below	Meet
shape_043	shape_035	Front	Meet
shape_044	shape_025	Above	Meet
shape_044	shape_029	Below	Meet
shape_U8	shape_024	Below	Overlap
shape_U8	shape_U6	Above	Overlap
shape_049	shape_010	Below	Meet
shape_049	shape_U2	Back	Meet
shape_058	shape_038	Front	Overlap
shape_058	shape_013	Right	Overlap

#### 5.2.2.10 Validación por expertos

En la implementación de este proceso, el experto identifica los elementos mal clasificados, con propiedades o relaciones incorrectas y los modifica libremente. El ciclo de vida del algoritmo termina cuando el experto da el visto bueno al modelo semántico presentado.



### 5.2.2.11 Resultados de la implementación del algoritmo

Finalmente se muestra el esquema en un grafo representado en la Figura 5.2.21, la cual muestra el gráfico final con las relaciones espaciales entre los segmentos que pertenecen a la nube de puntos.

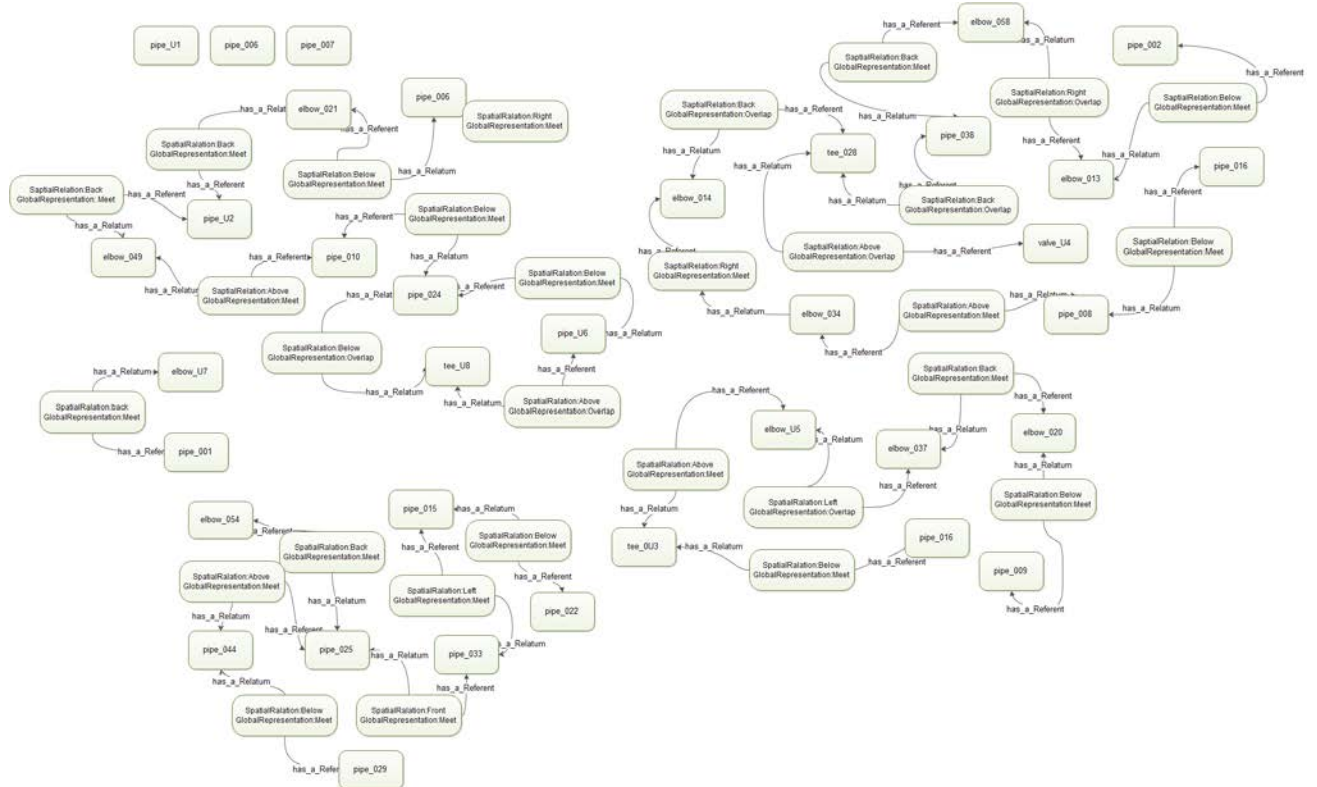


Fig. 5.2.21: Gráfico final

Gracias a la generación de información física, lógica y semántica a partir de una nube de puntos mediante la aplicación del algoritmo propuesto, es posible crear maquetas enriquecidas.

## 5.3 Validación semántica del algoritmo

En este punto, la maqueta digital enriquecida está compuesta de asociaciones físicas, lógicas y semánticas de los segmentos pertenecientes a la nube de puntos. El enriquecimiento de la información puede ser verificado dando respuesta a algunas consultas semánticas, por ejemplo: “¿Cuántos tubos con radio  $A$  existen en el la maqueta?”, “¿Qué elemento

está en la coordenada  $x, y, z$ ?”, “¿Cuál es el nombre del posible fabricante del tubo con centroide  $x, y, z$ ?”, tal como se muestra en los casos de prueba 1, 2 y 3, aplicados a la maqueta digital enriquecida obtenida en el apartado anterior.

### 5.3.1 Caso 1

#### El planteamiento

Se pretende dar respuesta a la pregunta *¿Qué elemento está en la coordenada  $x, y, z$ ?*, en donde  $x = -81.9557$ ,  $y = 14.2382$ ,  $z = 1.27649$ . Como resultado se espera una respuesta semántica y el objeto situado en la coordenada especificada dentro de la nube de puntos.

Tabla 5.3.1: Definición del caso 1

Consulta	<i>¿Qué elemento está en la coordenada <math>x, y, z</math>?</i>
Variables	$x = -81.9557$ , $y = 14.2382$ , $z = 1.27649$

#### La consulta

La consulta a la ontología propuesta se lleva a cabo filtrando aquellos elementos con centroide  $X = -81.9557$ ,  $Y = 14.2382$  y  $Z = 1.27649$ , tal y como se muestra en el script 2.

---

#### Script 2 Consulta a la ontología Caso 1

---

```
SELECT * WHERE {
  ?pipes rdf:type ont:Pipe .
  ?cs rdf:type ont:Centroid .
  ?pipes ont:has_a ?cs .
  ?cs ont:x ?xx .
  ?cs ont:y ?yy .
  ?cs ont:z ?zz .
  FILTER (?xx = -81.9557 && ?yy = 14.2382 && ?zz = 1.27649 )
}
```

---

#### El resultado

La información obtenida al crear la maqueta digital enriquecida, tiene la capacidad de dar respuesta a la pregunta *¿Qué elemento está en la coordenada  $x, y, z$ ?*. La consulta dentro de la ontología propuesta se desarrolla filtrando a un elemento que tiene un centroide con las coordenadas  $X = -81.9557$ ,  $Y = 14.2382$  y  $Z = 1.27649$ . La Figura 5.3.1 en a)

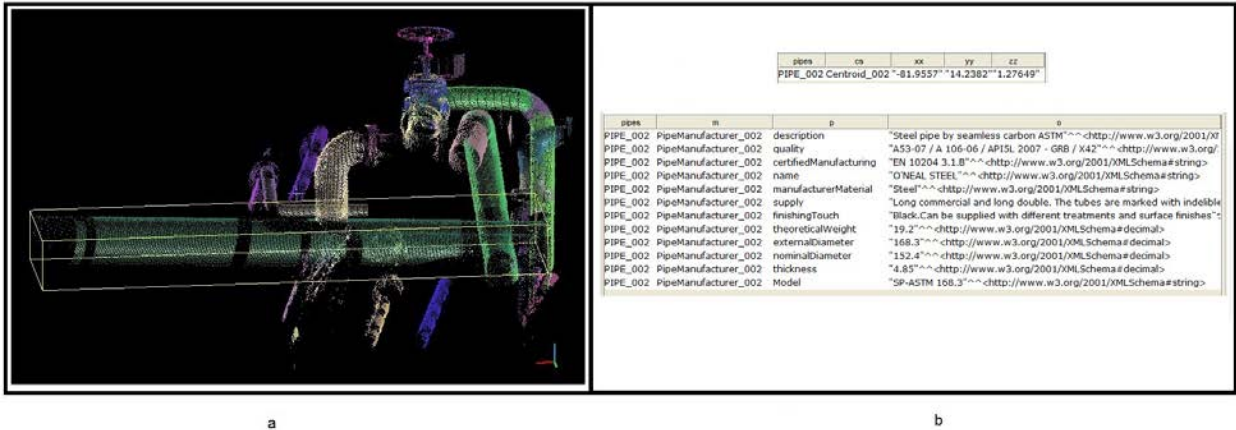


Fig. 5.3.1: a) Objeto dentro de la nube de puntos. b) Resultado semántico

muestra el elemento resultante marcado dentro de la nube de puntos, mientras que en b) se muestra el resultado semántico, es decir, elemento tipo *Pipe* llamado *Pipe\_002*, el cual tiene una propiedad diámetro de 152,4 mm. Con el valor de esta propiedad y el estandar de fabricación, en este caso *ASME B36.10* se puede sugerir al experto que el tubo en cuestión es un tubo de acero de ASTM de carbono sin costura, O'Neal Acero Fabricante, 168.3 Modelo ASTM-SP, con un acabado en negro. Se puede suministrar con diferentes tratamientos de superficie y acabados, tiene una fabricación certificada EN 10204 3.1.B y A53-07 calidad / A 106-06 / API5L 2007 - GRB / X42, entre otras características.

### 5.3.2 Caso 2

#### El planteamiento

Se pretende dar respuesta a la pregunta *¿Cuántos tubos con radio A existen en la maqueta?*, en donde  $A = 101.6$ . Como resultado se espera una respuesta semántica y el conjunto de objetos dentro de la nube de puntos que cumplan con la característica especificada.

Tabla 5.3.2: Descripción de caso 2

Consulta	¿Cuántos tubos con radio A existen en el la maqueta?
Variables	A= 101.6

#### La consulta

La consulta a la ontología propuesta se lleva a cabo filtrando los elementos por la propiedad diameter, donde  $diameter = 101.6$ , tal y como se muestra en el script 3.



**Script 3** Consulta a la ontología Caso 2

---

```

SELECT * {
  {
    SELECT * WHERE {
      ?pipes rdf:type ont:Pipe .
      ?pipes ont:diameter ?dd .
      FILTER (?dd = 101.6 )
    }
  }
  UNION {
    SELECT ( COUNT (*) as ?total)
    WHERE{
      ?pipes rdf:type ont:Pipe .
      ?pipes ont:diameter ?dd .
      FILTER (?dd = 101.6 )
    }
  }
}

```

---

**El resultado**

El resultado de esta consulta se muestra en la Figura 5.3.2, el cual representa, al igual que el caso anterior dos enfoques diferentes, en a) se presenta la respuesta semántica perteneciente a los objetos dentro con diámetro 101.6 mientras que en b) se observan los objetos referidos dentro de la nube de puntos.

El caso 2, pretende dar respuesta a la pregunta “¿Cuántos tubos con diámetro A existen en la maqueta?”, por lo que se hace una consulta a la ontología de todos los elementos tipo Pipe con un diámetro de 101,6, Figura 5.3.2 muestra en a) que el resultado semántico es una lista de 4 cilindros Pipe\_014, Pipe\_010, Pipe\_015 y Pipe\_004 y sus propiedades. Como en el caso 1, debido a que es un elemento tipo Pipe se relaciona con el catálogo de productores para conocer las características de mercado de los tubos, lo que sugiere que los tubos en cuestión son tubos de acero al carbono sin soldadura de ASTM, O’Neal Acero Fabricante, Modelo 114,3 ASTM SP-, que puede ser suministrado, siempre y comercial y largo doble. Los tubos están marcados con pintura indeleble y contener al menos la siguiente información: anagrama del fabricante, las dimensiones estándar y el número de calor, tener una fabricación certificada EN 10204 3.1.B y A53-07 calidad / A 106-06 / API5L 2007 - GRB / X42, entre otras características. Mientras que en b) se observan los objetos referidos dentro de la nube de puntos.

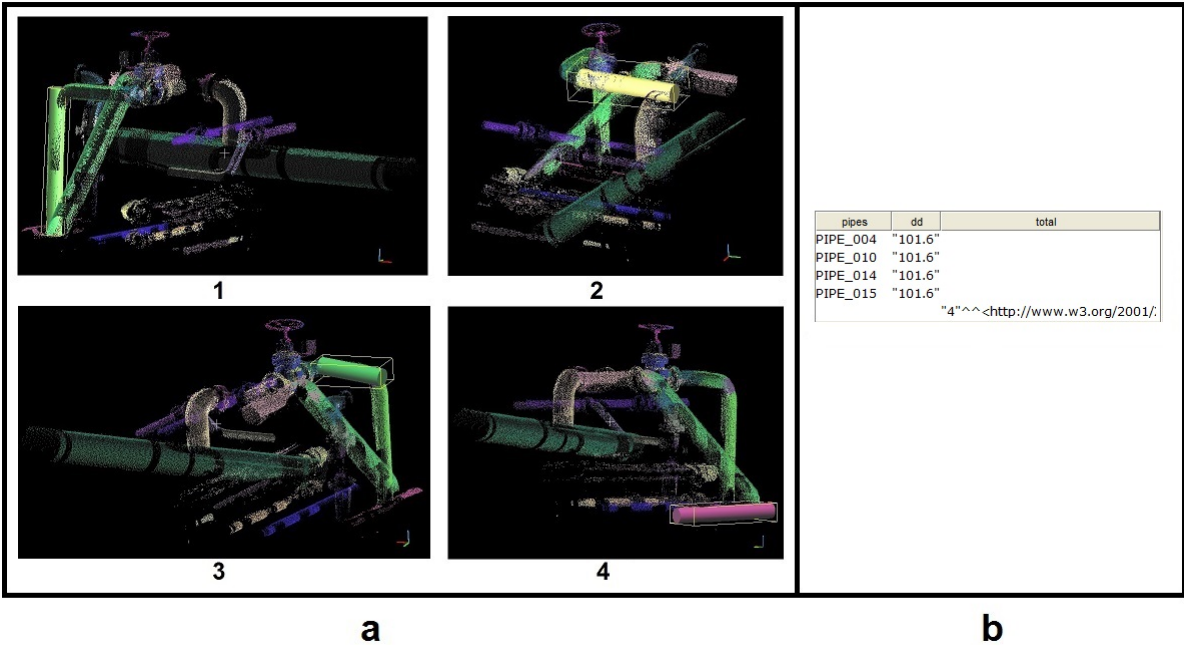


Fig. 5.3.2: a) Objetos dentro de la nube de puntos 1) Pipe\_004, 2) Pipe\_010, 3) Pipe\_014 y 4) Pipe\_015 b) Resultado semántico.

### 5.3.3 Caso 3

#### El planteamiento

Se pretende dar respuesta a la pregunta *¿Cuál es el nombre del posible fabricante del tubo con centroide  $x$ ,  $y$ ,  $z$ ?*, en donde  $x = -83.4685$ ,  $Y=13.331$ ,  $Z=1.4124$ . Como resultado se espera una respuesta semántica y el nombre del posible fabricante del elemento especificado.

Tabla 5.3.3: Descripción de caso 3

Consulta	<i>¿Cuál es el nombre del posible fabricante del tubo con centroide <math>x</math>, <math>y</math>, <math>z</math>?</i>
Variables	$x = -83.4685$ $Y=13.9215$ , $Z=1.4124$

#### La consulta

La consulta a la ontología propuesta se lleva a cabo filtrando los elementos pertenecientes a la nube de puntos que tengan un centroide en  $x= -83.4685$ ,  $y=13.331$ ,  $z=1.4124$ , tal y como se muestra en el script 4.

**Script 4** Consulta a la ontología Caso 3

---

```

1: SELECT * WHERE {
2:   ?pipes rdf:type ont:Pipe .
3:   ?cs rdf:type ont:Centroid .
4:   ?m rdf:type ont:PipeManufacturer .
5:   ?pipes ont:has_a ?cs .
6:   ?pipes ont:has_some ?m.
7:   ?cs ont:x ?xx .
8:   ?cs ont:y ?yy .
9:   ?cs ont:z ?zz .
10:  ?m ont:name ?mm.
11:  FILTER (?xx = -83.7301 && ?yy = 13.9215 && ?zz = 1.1515)
12: }

```

---

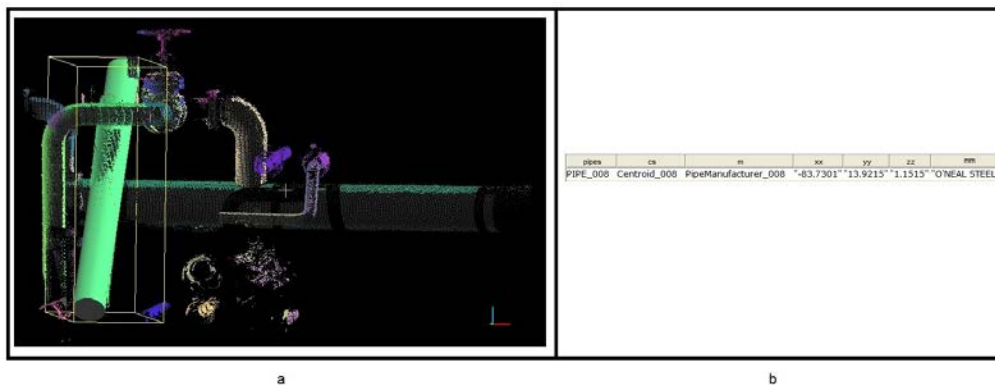
**El resultado**

Fig. 5.3.3: a

) Objeto dentro de la nube de puntos b) Resultado semántico.

El caso 3 tiene el objetivo de responder a la pregunta “¿Cuál es el nombre del posible fabricante del tubo con centroide x, y, z?”, Como se muestra en los ejemplos anteriores, se realiza una consulta de los elementos con un centroide  $X = -83.4685$   $Y = 13.331$ ,  $Z = 1.4124$ , obteniendo el nombre del posible fabricante del tubo seleccionado. El resultado semántico y el objeto dentro de la nube de puntos son presentados en la Figura 5.3.3.

## 5.4 Conclusiones de la validación

El objetivo del estudio experimental en comprobar las hipótesis H1, H2 y H3 de la presente tesis. En este capítulo se presentaron las tres etapas del estudio experimental (datos fuente, algoritmo inteligente y verificación) aplicado en un contexto industrial con el fin de implementar el algoritmo propuesto. Dicha implementación de algoritmo se llevó a cabo sobre una escena industrial real la cual, después de ser escaneada generó una nube de puntos que actuó como entrada al algoritmo propuesto y como salida se esperó una maqueta digital enriquecida, esto es, una descripción de la escena con información física, lógica y semántica.

El algoritmo fué implementado y descrito proceso a proceso:

1. El proceso de reconocimiento de primitivas. Compuesto por tres subprocesos: pre-proceso, extracción de descriptores y clasificación, en donde se eliminó el ruido de la nube de puntos original, se extrajeron descriptores visuales de primitivas geométricas definidas para este contexto (Cilindro, esfera y toro) y por último se implementó un algoritmo tradicional de reconocimiento de formas geométricas (HFP) seleccionado después de un completo análisis. El resultado de este proceso fue una lista de segmentos clasificados de acuerdo a su semejanza con las primitivas geométricas predefinidas con una etiqueta simple. La Tabla 5.2.6 muestra los segmentos y sus clasificaciones. Con la implementación de este proceso es posible responder la pregunta *¿Existe algún objeto conocido dentro de la escena escaneada?*, donde la respuesta es *"Sí, existen 57 objetos conocidos en la escena industrial, de los cuales 36 son Cilindros, 10 son Toros y 11 Esferas"*, esta información no es totalmente confiable debido a que presentan errores de bajo-segmentación y sobre-segmentación, éstos últimos solventados en el siguiente proceso.
2. Proceso de enriquecimiento semántico. Compuesto por cuatro subprocesos: cálculo de características geométricas, cálculo de la topología. Los cuales se apoyan en un modelo semántico conformado con una ontología y reglas semánticas. Se desarrolló una ontología personalizada para el caso de uso industrial, creada a partir de la metodología Methodology. La ontología propuesta describe las relaciones y el comportamiento de los elementos comunes en una escena industrial, para asegurar ese comportamiento se crearon reglas semánticas. El cálculo de las características geométricas se hace de acuerdo a la clasificación asignada al segmento, se calcula información como longitud, radio, circunferencia, etc., según corresponda. Mientras que el cálculo de la topología se lleva a cabo implementando reglas semánticas, las cuales ayudan también a corregir errores de sobre-segmentación. El resultado de este proceso fue una escena digital descrita detalladamente con información lógica, física y semántica. Con la implementación de este proceso es posible responder la pregunta *¿Qué tipo de elementos tiene la escena escaneada?*, donde la respuesta se puede observar en la Figura 5.2.21, la cual presenta el grafo generado con la descripción lógica

física y semántica de la escena digital.

Además presentaron validaciones semánticas sobre el modelo semántico final generado en este caso de uso. A partir de la implementación y validación del caso de uso es posible comprobar la **H1**: *Es posible mejorar el reconocimiento de objetos a partir de la acotación semántica de primitivas que describan elementos comunes dentro de una escena industrial, específicamente cilindros, toros y esferas pertenecientes a una nube de puntos 3D considerando la combinación de descriptores de características geométricas y el uso de reglas semánticas.* Se ha comprobado la hipótesis debido a que con un algoritmo de reconocimiento de figuras geométricas tradicional no es posible obtener una descripción semántica de los elementos ni una descripción topológica de la misma. La **H2**: *Es posible enriquecer una nube de puntos detallando la definición de sus objetos de acuerdo al cálculo de sus características a partir de descriptores visuales.* Se ha comprobado la hipótesis debido al uso de descriptores visuales como base para la clasificación inicial de los segmentos y después calculando información lógica, física y semántica de cada uno. Y la **H3**: *La información proporcionada por este algoritmo puede aplicarse al reconocimiento de un entorno real.* Se ha comprobado debido a la implementación del algoritmo en un entorno industrial real.



En este capítulo tiene como objetivo evaluar el algoritmo propuesto en esta tesis. Realizar evaluaciones a algoritmos de reconocimiento de primitivas geométricas no es una tarea trivial, debido a los diversos enfoques que pueden ser medidos y evaluados, en donde, criterios de evaluación deben ser establecidos de acuerdo al objetivo final.

### 6.1 Enfoques de evaluación

Desde una perspectiva general, se pueden mencionar dos enfoques diferentes para evaluar software y herramientas: métodos cuantitativos y cualitativos, los cuales se describen brevemente a continuación.

#### 6.1.1 Métodos cuantitativos de evaluación.

Los métodos cuantitativos se basan en el supuesto de que el producto de software tiene al menos una propiedad medible, la cual puede cambiar de acuerdo a la naturaleza de los métodos o herramientas que son evaluados.

Los valores cuantificables pueden ser frecuencias, porcentajes, costos, tasas y magnitudes, entre otros. La información recopilada a través de estos métodos permite:

- Reportar sobre indicadores definidos
- Reportar comparativas de características específicas entre herramientas.

Las evaluaciones cuantitativas pueden desarrollarse en tres formas diferentes:

- Casos de estudio. Tienen el objetivo de analizar un contexto en específico donde se implementa el método o herramienta. Identifican y documentan características específicas que pueden afectar los resultados de una actividad.
- Experimentos formales. Un experimento es un estudio en el que se manipulan intencionalmente una o más variables independientes y se analizan las consecuencias que la manipulación tiene sobre una o más variables dependientes. Los experimentos formales tienen el objetivo de analizar cuantitativamente de manera controlada y rigurosa aquellos factores que afectan a las herramientas o métodos evaluados.
- Encuestas. Son estudios retrospectivos con el objetivo de documentar relaciones y resultados. Consiste en un conjunto predeterminado de preguntas normalizadas, que recopila información directamente de una muestra seleccionada, como resultado es posible obtener datos estadísticos sobre opiniones, hechos u otras variables.

Por otra parte, de acuerdo al contexto de esta tesis, es preciso analizar evaluaciones de sistemas de clasificación, los cuales se basan en la medición de la predicción. Dicha evaluación tiene el objetivo de medir la capacidad de representación, es decir la manera en que las reglas generadas para la clasificación reproducen con exactitud el comportamiento de los elementos en el mundo real; y generalización, es decir cómo las reglas generadas predicen patrones nuevos.

Para realizar medidas numéricas de la capacidad de representación y de generalización se utiliza como referencia un sistema de clasificación con dos clases: P (Positiva) y N (Negativa). Después, se define y calcula una *matriz de confusión*, que se usa para comparar el número de elementos que el clasificador predice para cada una de las clases, con el valor esperado, es decir el valor real. En la Tabla 6.1.1, los “True Positives” (TP) y “True Negatives” (TN) se refieren a elementos correctamente clasificados, mientras que los “False Positives” (FP) y “False Negatives” (FN) son elementos clasificados como de la clase incorrecta.

A partir de la matriz de confusión es posible evaluar de diferentes formas la calidad de un clasificador. El índice más utilizado en la literatura es la tasa de éxito de la clasificación (*accuracy* en inglés), ésta puede ser calculada mediante 6.1.1.

$$Tasa\ de\ éxito = \frac{TP + TN}{TP + TN + FP + FN} \quad (6.1.1)$$

Tabla 6.1.1: Matriz de confusión

		Clase Esperada	
		P	N
Predicción	P	TP	FP
	N	FN	TN



La tasa de éxito se usa comúnmente como una medida global para evaluar un sistema de clasificación completo, por lo que para evaluaciones parciales, se utilizan expresiones que hacen referencia al conjunto de elementos clasificados por las reglas. La precisión, sensibilidad y especificidad son mas medidas más utilizadas.

La medida Precisión (*precision*) considera solo los elementos clasificados en una clase, por lo que es una medida ideal para conocer la calidad de clasificación por clase. La precisión se calcula a través de 6.1.2.

$$P = \frac{TP}{TP + FP} \quad (6.1.2)$$

Por otra parte la sensibilidad (*sensitivity* o *recall*) actúa como medida complementaria a la precisión. La sensibilidad es una forma de evaluar la “cobertura” del clasificador, es decir se calcula en qué medida la regla es suficientemente “sensible” para detectar los elementos de la clase P. La sensibilidad se calcula por 6.1.3.

$$R = \frac{TP}{TP + FN} \quad (6.1.3)$$

Finalmente la Especificidad es una medida análoga a la precisión, pero considerando los elementos de la clase N. La especificidad se calcula mediante 6.1.4.

$$E = \frac{TN}{TN + FP} \quad (6.1.4)$$

Una forma más exacta de medir la calidad de una regla que forma parte de un clasificador, es encontrando la relación entre precisión y sensibilidad, esta no es una tarea trivial debido a que los objetivos de ambas medidas suelen estar en conflicto. Es decir, en un caso trivial en que existan tantas reglas como elementos, y cada regla asigna exactamente la clase esperada a su elemento, la precisión será 1.0 esto es máxima precisión; sin embargo, la sensibilidad de cada una de estas reglas es mínima. En el extremo contrario del espectro, se encuentra el caso donde una regla asigna una de las clases a todos los elementos existentes, esto significa que la sensibilidad para los elementos de dicha clase es 1.0, por lo tanto sensibilidad máxima, pero la precisión será mínima, puesto que clasifica el resto de los elementos como falsos positivos. Por lo que, una forma convencional de solucionar este problema es agregar ambas medidas en una única función. Esta aproximación puede verse solucionada mediante el análisis de tablas ROC. Una tabla ROC es un diagrama en dos dimensiones en el que se representa en el eje Y la tasa de verdaderos positivos y en el eje X la tasa de falsos negativos. Estas tasas se pueden calcular a través de 6.1.5 y 6.1.6 respectivamente.

$$Tasa_{tp} = \frac{TP}{TP + FN} \quad (6.1.5)$$

$$Tasa_{fp} = \frac{FP}{TP + FN} \quad (6.1.6)$$

Un ejemplo de evaluación cuantitativa de métodos o herramientas de software es la presentada en H. Li et al., 2015, donde fueron descritos experimentos para validar la eficacia del descriptor QDFT en la transformación geométrica y los indicadores a medir fueron la robustez frente al ruido de dicho descriptor, así como una comparación de rendimiento en el volumen de datos de gran tamaño. Por otra parte, experimentos de clasificación multiclase uno-versus-resto se abordan por C. Hong et al., 2015, donde las muestras verdaderas y falsas fueron etiquetados de forma diferente. Los autores utilizaron índices de precisión para evaluar el rendimiento de diferentes métodos. De la misma forma, en Rubio et al., 2015 propuso una comparación de rendimiento entre su investigación y otros siete estudios similares, midiendo la precisión media para cada uno. Por último, los autores en Ruiz-Sarmiento et al., 2015 el éxito de reconocimiento de objetos de la investigación fue discutido, midiendo diferentes aspectos tales como la inclusión de información contextual sobre los objetos, sus características geométricas y apariencia, y su clasificación en función de su tipo. Las métricas utilizadas fueron precisión y la recuperación.

### 6.1.2 Los métodos cualitativos

Los métodos cualitativos de evaluación permiten recolectar información que proporcionan resultados que no pueden medirse fácilmente o transferirse a números.

El término análisis de funciones es utilizado en la literatura para describir una evaluación cualitativa, la cual se basa en:

1. La identificación de las necesidades que tienen los usuarios para una tarea o actividad en particular y
2. El mapeo de esos requisitos a las características que al método o herramienta destinado a apoyar esa tarea o actividad debería poseer.

Los métodos cualitativos pueden incluir actividades como pedir a las personas responder preguntas de “ensayo” acerca de asuntos complejos u observar interacciones en situaciones complejas.

Un ejemplo de una evaluación cualitativa es la comparación proporcionada por Attene, Falcidieno, et al., 2006, donde se analizaron dos métodos los cuales tenían el objetivo de segmentar una nube de puntos. Los dos métodos fueron ejecutados sobre la misma nube de puntos para determinar cuál de ellos lograba generar el mejor modelo estilizado.

### 6.1.3 Discusión de los enfoques de evaluación

Como ya se ha comentado, existen dos tipos de enfoques de evaluación de software: cuantitativos y cualitativos. Los primeros asumen que el software tiene al menos una propiedad medible y se basa en procesos matemáticos para obtener una puntuación de la propiedad seleccionada. Por otra parte los cualitativos que suelen analizar información y a partir de

ella obtener resultados o conclusiones. Se selecciona la evaluación cuantitativa debido a que se pretende medir la calidad de la clasificación del algoritmo, para que de esta manera se pueda comprobar la H1 de la presente tesis.

## 6.2 Composición de la prueba

Los criterios de evaluación seleccionados son:

1. Segmentos clasificados. Número de elementos segmentados pertenecientes a la nube de puntos original.
2. Segmentos clasificados correctamente. Se refiere a los segmentos clasificados correctamente de acuerdo a sus propiedades geométricas es decir, un Verdadero Positivo (TP).
3. Falsos positivos. Representan a los segmentos clasificados en una clase que no corresponde a sus propiedades geométricas reales (FP).
4. Falsos negativos. Son los segmentos pertenecientes a una clase y que el algoritmo clasificó en otra (FN).
5. Precisión. Se refiere a la fracción de segmentos clasificados y que son relevantes para el proceso (P). General mente la precisión se define como la proximidad entre las indicaciones o valores medidos de un mismo mensurando, obtenidos en mediciones repetidas, bajo condiciones especificadas. Cuanto más estrecha sea la distribución de resultados, menor será la desviación típica de la misma y mayor la precisión de la medida.
6. Recall. Indica la fracción de los segmentos relevantes para el proceso, pero que también fueron clasificados correctamente (R).
7. Exactitud. (A) se define como la proximidad entre el valor medido y el valor “verdadero” del mensurando. Por lo que, una medición es más exacta cuanto más pequeño es el error de medida.
8. Sobre segmentación. (Over-segmentation -OS-) Es el proceso por el cual los objetos que se están segmentados son a su vez segmentado o fracturado en subcomponentes. En esta evaluación lo utilizaremos como un indicador de calidad.

La metodología propuesta para evaluar la calidad de segmentación del algoritmo propuesto en esta tesis se muestra en la Figura 6.2.1, la cual consiste en 1) Seleccionar una nube de puntos, 2) esta nube de puntos será procesada y analizada por el algoritmo tradicional de reconocimiento de formas geométricas llamado RansacSD, 3) a partir de los resultados

obtenidos se medirá la calidad de clasificación de los mismos, 4) esta misma nube de puntos se procesará y analizará por el algoritmo propuesto por esta tesis, 5) se evaluarán la calidad de sus resultados calculando los criterios de evaluación descritos anteriormente, y 6) se realizará una comparativa de las medidas de calidad con el fin de conocer que algoritmo tiene mayor calidad en la clasificación de formas geométricas.

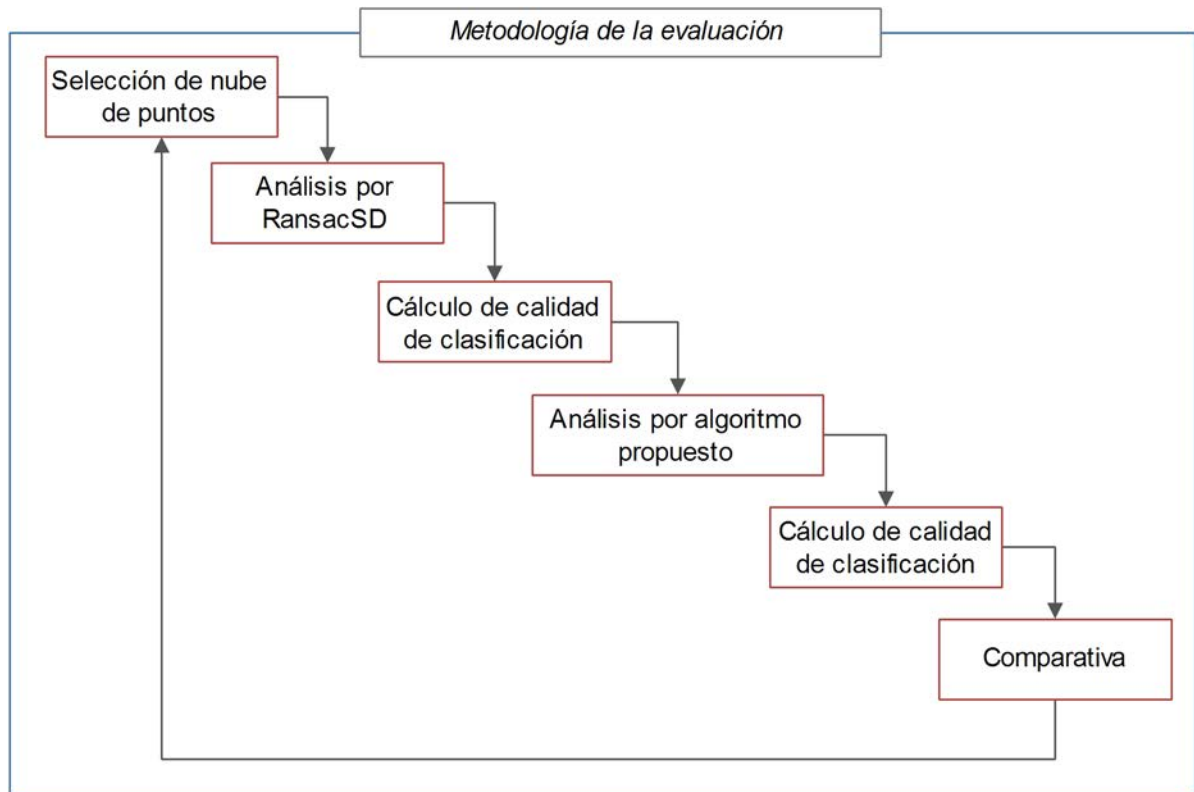


Fig. 6.2.1: Metodología de la evaluación

### 6.3 Definición de datos reales

Esta metodología se aplicó sobre cinco nubes de puntos de escenas industriales reales, las cuales se muestran en las figuras 6.3.1, 6.3.2, 6.3.3, 6.3.4 y 6.3.5. Las características reales de las nubes de puntos son descritas en la Tabla 6.3.1, donde se puede observar que cada nube de puntos se identifica por medio de un ID, las nubes de puntos fueron seleccionadas a modo que se pudiera evaluar el algoritmo contemplando diferentes umbrales, es decir, diferentes tamaños de nubes de puntos y diferentes formas geométricas a reconocer. El tamaño de las nubes de puntos van desde 60955 hasta 176963 puntos, mientras que las

formas geométricas a clasificar en estas pruebas son cilindros, planos y toros, tal y como muestra la Tabla 6.3.1.

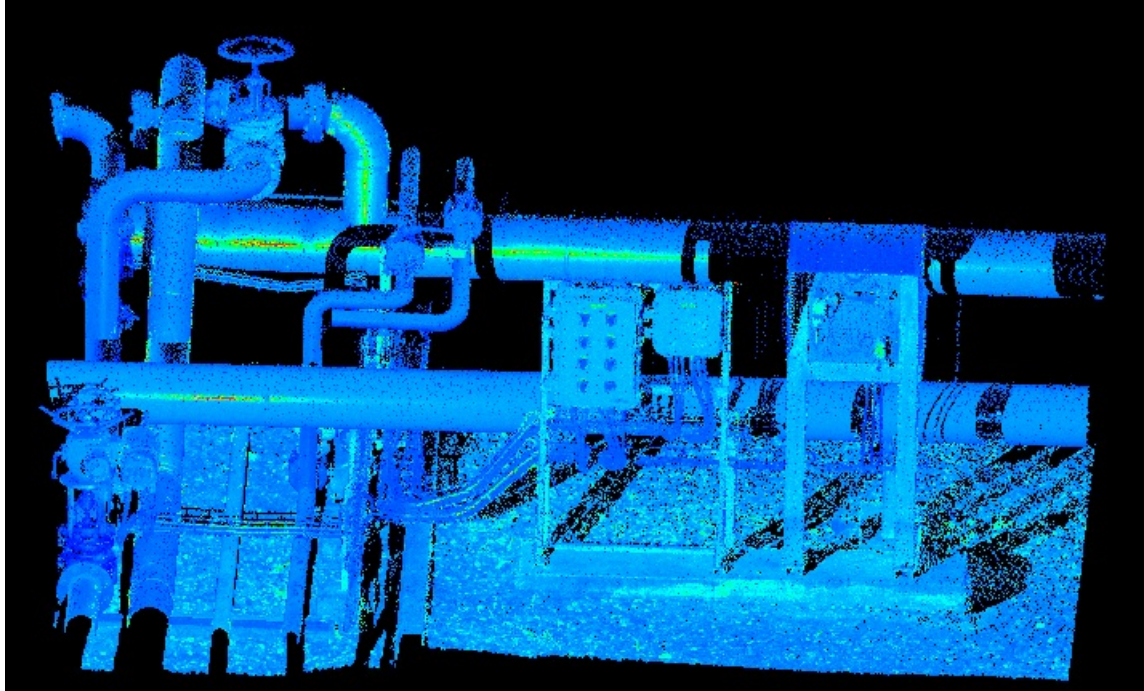


Fig. 6.3.1: Test 1. Nube de puntos de escena industrial

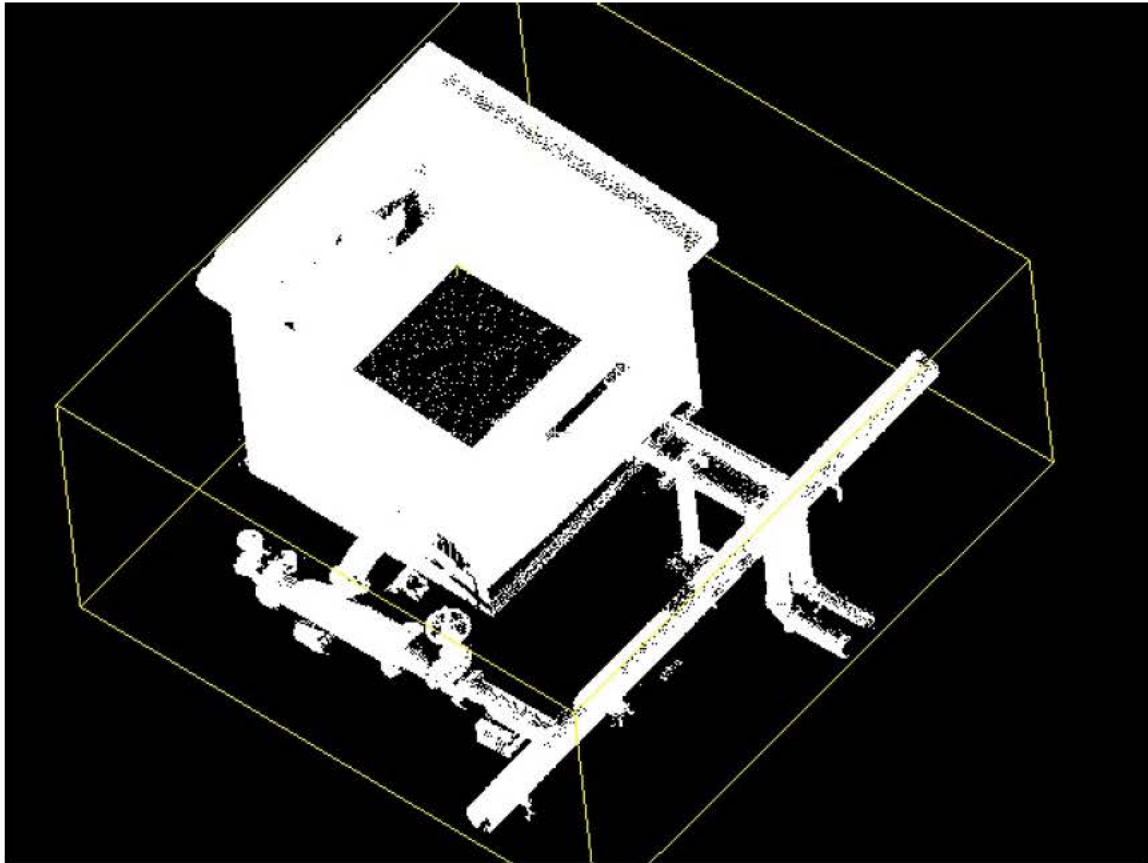


Fig. 6.3.2: Test 2. Nube de puntos de escena industrial

Tabla 6.3.1: Información real de nubes de puntos a evaluar

<i>Cloud Point ID</i>	<i>Num de puntos</i>	<i>Num Items</i>	<i>Formas geométricas a reconocer</i>
1	603444	80	Cilindros y planos
2	1776963	122	Cilindros, planos y toros
3	239816	45	Cilindros y planos
4	60955	11	Cilindros y planos
5	603444	83	Cilindros, planos y toros

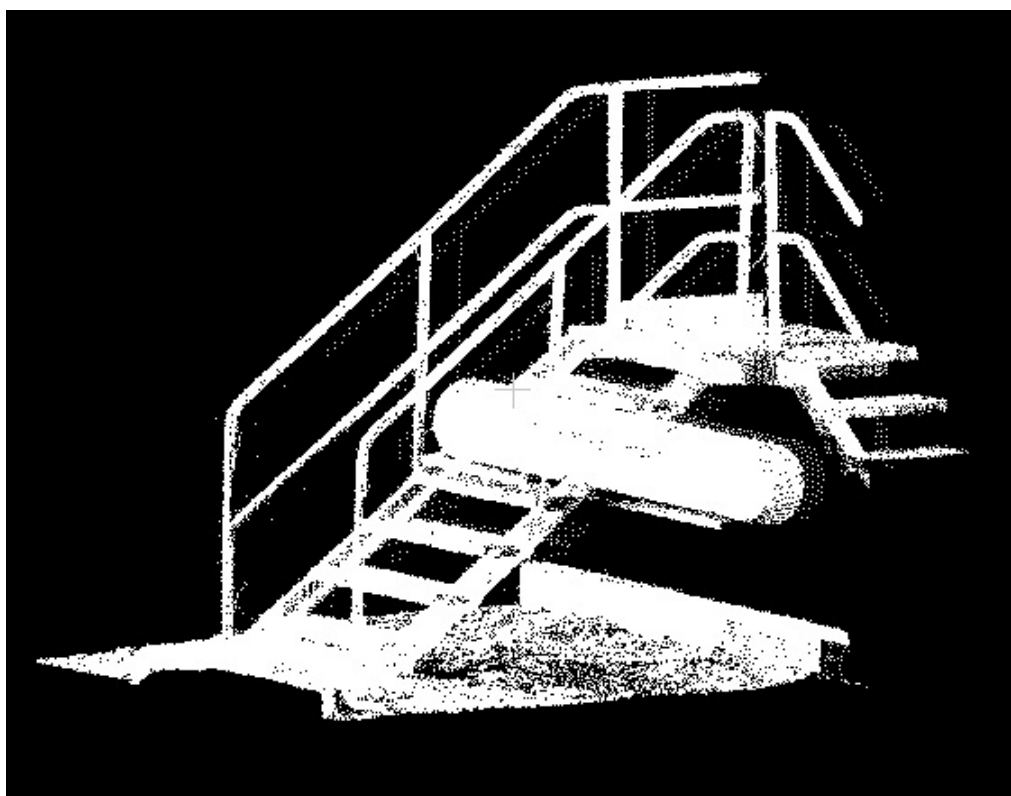


Fig. 6.3.3: Test 3. Nube de puntos de escena industrial

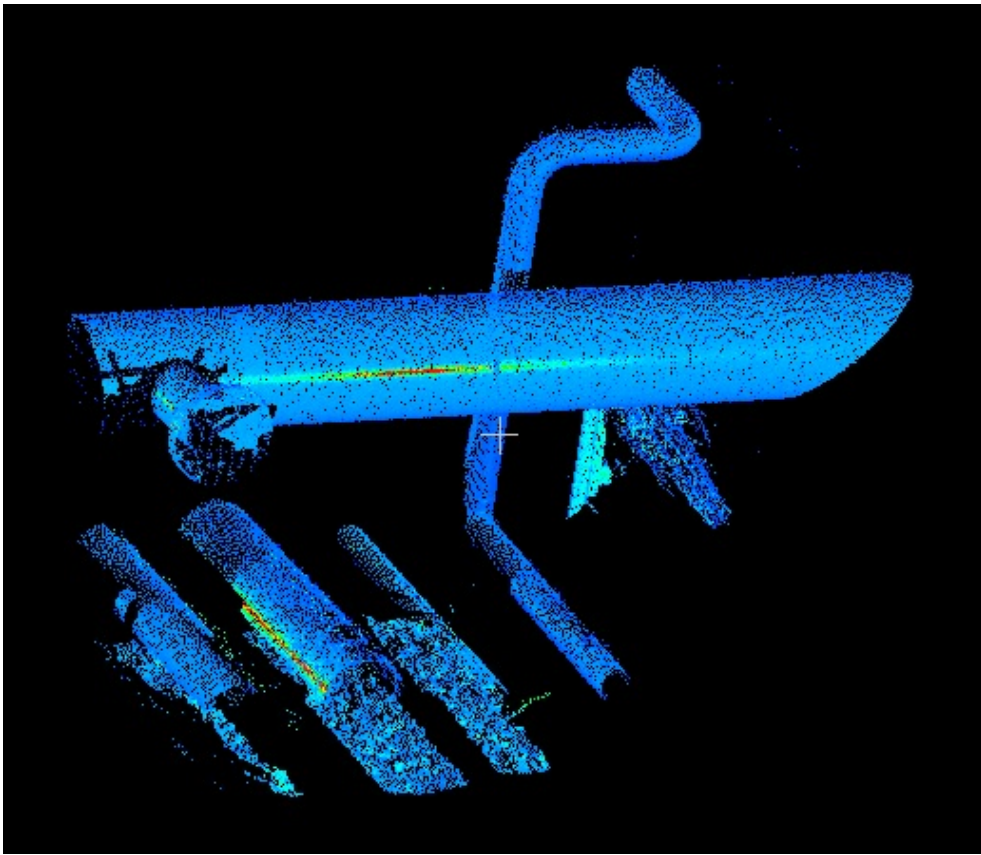


Fig. 6.3.4: Test 4. Nube de puntos de escena industrial



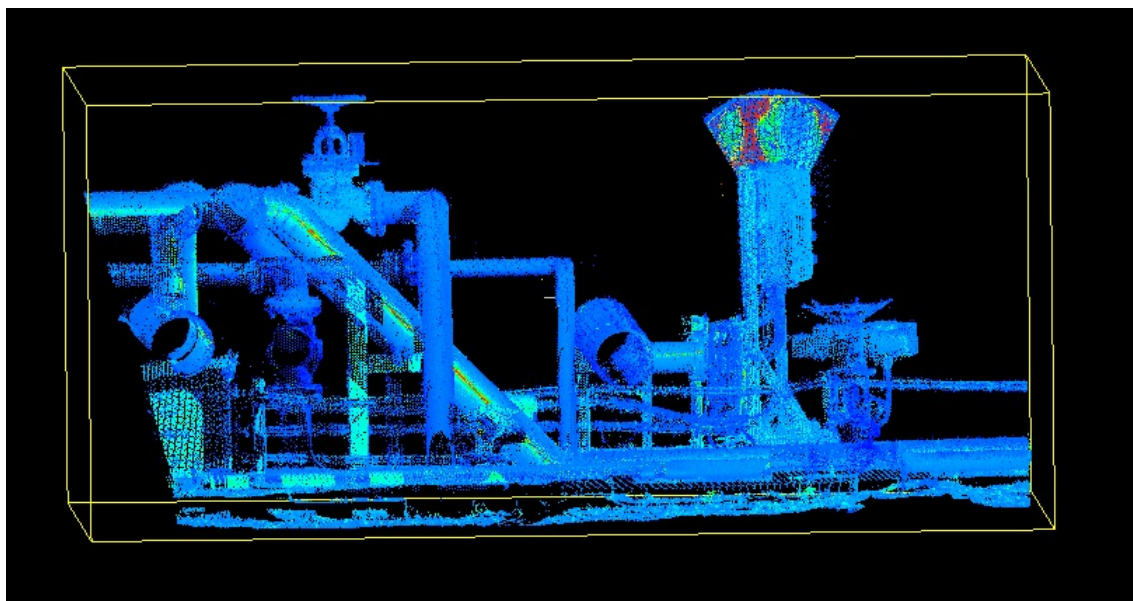


Fig. 6.3.5: Test 5. Nube de puntos de escena industrial

## 6.4 Resultados de la evaluación

Antes de analizar los resultados de la evaluación es preciso recordar que uno de los objetivos principales del algoritmo es describir semánticamente los elementos que conforman una nube de puntos de una escena industrial, para lo cual la calidad de la segmentación o clasificación es un factor clave para este reto.

Tabla 6.4.1: Plantilla de matriz de confusión

<i>[Elemento a analizar]</i>	<i>[ID de nube de puntos]</i>	
	Valor de TP	Valor de FP
	Valor de FN	Valor de TN

Los datos se muestran de acuerdo al orden de la metodología utilizada en esta evaluación. Se utilizan matrices de confusión para presentar los resultados obtenidos por el algoritmo propuesto, la Figura 6.4.1 muestra la plantilla a utilizar para analizar los resultados, la cual identifica en la cabecera el Id de la nube de puntos a la que hace referencia la prueba, mientras que en la columna de la izquierda mostrará el elemento evaluado (*Pipe*, *Plane* o *Elbow*). Las casillas de intersección muestran los valores obtenidos en la prueba para TP, FP, FN y TN en estricto orden.

Tabla 6.4.2: Matrices de confusión de RansacSD

	ID = 1		ID = 2		ID = 3		ID = 4		ID = 5	
Pipe	37	18	23	58	26	1	8	1	36	11
	0	25	1	40	0	18	0	2	0	36
Plane	20	5	33	6	7	11	2	0	25	3
	6	49	57	26	1	26	0	9	2	53
Elbow			2	0					7	1
			0	120					0	75

Inicialmente se presenta en la Tabla 6.4.2 las matrices de confusión de RansacSD para cada una de las pruebas y con respecto a cada tipo de forma geométrica evaluada. Con el fin de calcular el ROC, para cada una de estas matrices se calcularon las tasas de verdaderos positivos y las tasas de falsos negativos (ver Fórmulas 6.1.5). La Tabla 6.4.3 muestra las tasas calculadas, la cuales servirán de fuente de datos para dibujar el ROC.

Tabla 6.4.3: Cálculo de tasas TP y FP de RansacSD

ID de experimento	Cloud Point ID	Shape	FN	FP	TP	TN	Tasa TP = $\frac{TP}{TP+FN}$	Tasa FP = $\frac{FP}{FP+TN}$
1_Pipe	1	Pipe	0	18	37	25	1	0.33
1_Plane	1	Plane	6	5	20	49	0.77	0.20
2_Pipe	2	Pipe	1	58	23	40	0.96	0.72
2_Plane	2	Plane	57	6	33	26	0.37	0.15
2_Elbow	2	Elbow	0	0	2	120	1	0
3_Pipe	3	Pipe	0	1	26	18	1	0.04
3_Plane	3	Plane	1	11	7	26	0.88	0.61
4_Pipe	4	Pipe	0	1	8	2	1	0.11
4_Plane	4	Plane	0	0	2	9	1	0
5_Pipe	5	Pipe	0	11	36	36	1	0.23
5_Plane	5	Plane	2	3	25	53	0.93	0.11
5_Elbow	5	Elbow	0	1	7	75	1	0.13

En la Figura 6.4.1 se puede ver la dispersión de los puntos a través de la gráfica, recordemos que un espacio ROC se define por la tasa de verdaderos positivos y la tasa de falsos positivos como ejes x e y respectivamente, y representa los intercambios entre sensibilidad y 1-especificidad. Cada resultado de predicción o instancia de la matriz de confusión representa un punto en el espacio ROC.

El mejor método posible de predicción, es decir una clasificación perfecta, se situaría en un punto en la esquina superior izquierda, o coordenada (0,1) del espacio ROC, representando un 100% de sensibilidad y un 100% también de especificidad.

Por otra parte, una clasificación totalmente aleatoria daría un punto a lo largo de la línea diagonal (también llamada línea de no-discriminación), la cual divide el espacio ROC y va desde el extremo inferior izquierdo hasta la esquina superior derecha, se puede observar esta línea de no-discriminación en rojo en las figuras 6.4.1 y 6.4.2. Los puntos que caen por encima de la diagonal representan los buenos resultados de clasificación, a diferencia de los que se distribuyen por debajo, considerados como mala clasificación.

Como podemos observar, el algoritmo tradicional de reconocimiento de formas RansacSD presenta una calidad de clasificación buena, pero con ciertas dispersiones.

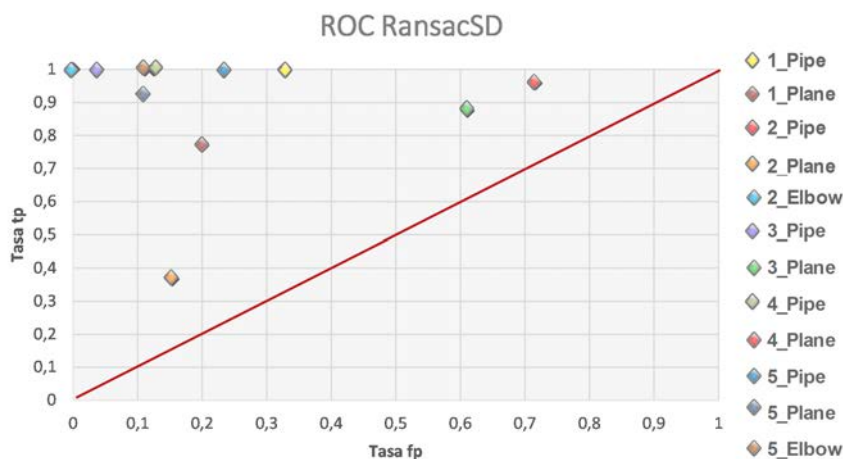


Fig. 6.4.1: ROC RansacSD

A continuación se analizan los resultados de aplicar el algoritmo sugerido en esta tesis a las mismas nubes de puntos.

Tabla 6.4.4: Matrices de confusión de Algoritmo propuesto

	ID = 1		ID = 2		ID = 3		ID = 4		ID = 5	
Pipe	31	12	19	1	16	0	7	1	26	9
	0	21	0	42	0	19	0	2	0	22
Plane	16	5	37	5	8	11	2	0	12	3
	0	43	0	20	0	16	0	8	0	42
Elbow			2	0					6	1
			0	60					0	50

La Tabla 6.4.4 muestra las matrices de confusión de los resultados obtenidos por el algoritmo propuesto. De la misma forma que RansacSD, a partir de la matriz de confusión de calcularon las tasas de TP y FP, las cuales se muestran en la Tabla 6.4.5. A partir de esta información es posible calcular en ROC de dicho algoritmo, el cual se presenta en la Figura 6.4.2. Como se puede observar los puntos se aproximan a la esquina superior izquierda, lo que se considera como una clasificación de calidad alta. Y mejoran al ROC que presentó el algoritmo tradicional RansacSD.

Tabla 6.4.5: Cálculo de tasas TP y FP del Algoritmo propuesto

ID de experimento	Cloud Point ID	Shape	FN	FP	TP	TN	Tasa TP = $\frac{TP}{TP+FN}$	Tasa FP = $\frac{FP}{FP+TN}$
1_Pipe	1	Pipe	0	12	31	21	1	0.28
1_Plane	1	Plane	0	5	16	43	1	0.24
2_Pipe	2	Pipe	0	1	19	42	1	0.05
2_Plane	2	Plane	0	0	37	20	1	0.12
2_Elbow	2	Elbow	0	0	2	60	1	0
3_Pipe	3	Pipe	0	0	16	19	1	0
3_Plane	3	Plane	0	11	8	16	1	0.58
4_Pipe	4	Pipe	0	1	7	2	1	0.13
4_Plane	4	Plane	0	0	2	8	1	0
5_Pipe	5	Pipe	0	9	26	22	1	0.26
5_Plane	5	Plane	0	3	12	42	1	0.20
5_Elbow	5	Elbow	0	1	6	50	1	0.14

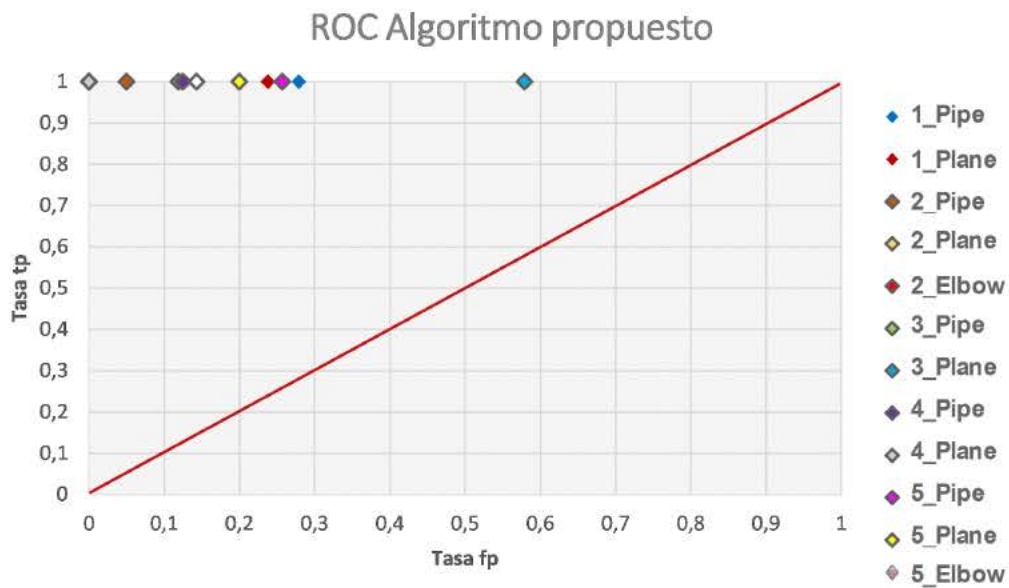


Fig. 6.4.2: ROC Algoritmo propuesto

En una vista más general de los resultados se presenta una comparativa de ambos

algoritmos. La Tabla 6.4.6 muestra una comparativa de la implementación del algoritmo RansacSD y el algoritmo propuesto sobre cinco nubes de puntos. La primera columna identifica la nube de puntos evaluada por medio de su ID, la segunda columna muestra el algoritmo ejecutado (RansacSD o el propuesto en esta tesis). La tercera columna presenta el número de elementos (*Pipes*, *Plane* o *Elbow*) pertenecientes a la nube de puntos de analizada. La cuarta columna muestra el tipo de elemento. A partir de esta columna se muestran los resultados de las pruebas, se presentan el número de FN, FP, TP y TN obtenidos, además de los porcentajes de P, R, A, OS.

Se puede observar que a pesar de que RansacSD obtiene resultados aceptables con precisiones de hasta 100% en caso de elementos que se asemejan a codos y hasta 96% en el reconocimiento de cilindros, tiene grandes problemas de sobre-segmentación. A diferencia del algoritmo propuesto, donde estos problemas son solventados alcanzando un 0%. Se puede observar en la reducción del número total de elementos en la columna 3 de la Tabla 6.4.6 para el algoritmo sugerido en esta tesis, donde dicha diferencia se debe a la unión de los elementos sobre-segmentados llevado a cabo por medio de la implementación de las reglas semánticas propuestas.



Tabla 6.4.6: Resultado de la evaluación

<i>Cloud Point ID</i>	<i>Algoritmo</i>	<i>Num Items</i>	<i>Shape</i>	<i>FN</i>	<i>FP</i>	<i>TP</i>	<i>TN</i>	<i>P</i>	<i>R</i>	<i>A</i>	<i>O-S</i>
1	RansacSD	80	Pipe	0	18	37	25	67%	100%	78%	16%
			Plane	6	5	20	49	80%	77%	86%	65%
	Propuesto	64	Pipe	0	12	31	21	72%	100%	81%	0%
			Plane	0	5	16	43	76%	100%	92%	0%
3	RansacSD	122	Pipe	1	58	23	40	28%	96%	52%	52%
			Plane	57	6	33	26	85%	37%	48%	67%
			Elbow	0	0	2	120	100%	100%	100%	0%
	Propuesto	62	Pipe	0	1	19	42	95%	100%	98%	0%
			Plane	0	5	37	20	88%	100%	92%	0%
			Elbow	0	0	2	60	100%	100%	100%	0%
4	RansacSD	45	Pipe	0	1	26	18	96%	100%	98%	62%
			Plane	1	11	7	26	39%	88%	73%	43%
	Propuesto	35	Pipe	0	0	16	19	100%	100%	100%	0%
			Plane	0	11	8	16	42%	100%	69%	0%
5	RansacSD	11	Pipe	0	1	8	2	89%	100%	91%	25%
			Plane	0	0	2	9	100%	100%	100%	0%
	Propuesto	10	Pipe	0	1	7	2	88%	100%	90%	0%
			Plane	0	0	2	8	100%	100%	100%	0%
7	RansacSD	83	Pipe	0	11	36	36	77%	100%	87%	33%
			Plane	2	3	25	53	89%	93%	94%	76%
			Elbow	0	1	7	75	88%	100%	99%	29%
	Propuesto	57	Pipe	0	9	26	22	74%	100%	84%	0%
			Plane	0	3	12	42	80%	100%	95%	0%
			Elbow	0	1	6	50	86%	100%	98%	0%

## 6.5 Conclusiones de la evaluación

Se han analizado los diferentes enfoques existentes para la evaluación del algoritmo: cualitativo y cuantitativo. Se eligió una evaluación cuantitativa debido a la necesidad de medir una propiedad del algoritmo, esto es, la calidad de la clasificación. Se utilizó una metodología para el desarrollo de dicha evaluación tal y como lo muestra la Figura 6.2.1:

1. Selección de la nube de puntos. Se evaluaron cinco nubes de puntos con características diferentes, de modo que los algoritmos pudieran ser evaluados en distintos entornos y las pruebas fueran imparciales.
2. Análisis por RansacSD. Las cinco nubes de puntos fueron analizadas por el algoritmo RansacSD.
3. Cálculo de la calidad de la clasificación. La Figura 6.4.1 muestra que los resultados obtenidos por RansacSD en la clasificación, tal y como se observa en el espacio ROC, los puntos están distribuidos por el espacio superior, por lo que se puede concluir que para la muestra evaluada, RansacSD es un buen clasificador. Presenta fallas tanto falsos positivos como falsos negativos y problemas de sobre-segmentación y bajo-segmentación.
4. Análisis de algoritmo propuesto. Las cinco nubes de puntos fueron analizadas por el algoritmo propuesto.
5. Cálculo de la calidad de la clasificación. La Figura 6.4.2 muestra que los resultados obtenidos por el algoritmo propuesto en esta tesis, se observa que los puntos tienden a 1 en el eje de las  $y$  (Tasa de TP) dentro del espacio ROC, esto quiere decir que no solo están distribuidos en el espacio ROC superior, sino que la calidad de la clasificación es mayor a la presentada por RansacSD. El algoritmo propuesto tiene a 0 en los falsos negativos y elimina los problemas de sobre-segmentación. Sin embargo, presenta fallas de falsos positivos y problemas de bajo-segmentación.
6. Comparativa. Se presentó la Tabla 6.4.6 donde se puede observar la diferencia entre los resultados de RansacSD y el algoritmo propuesto, para cada una de las nubes de puntos y formas geométricas reconocidas.

Como resultado final de este apartado, podemos concluir que el algoritmo propuesto obtiene mejores resultados que la clasificación de un algoritmo tradicional de acuerdo con los resultados del paso 6 donde, para el algoritmo propuesto, los porcentajes de sobre segmentación (OS) bajaron hasta un 0% y tanto los porcentajes de Recall (R), Precisión (P), como el de Exactitud (A) aumentaron.



---

### Conclusiones y trabajo Futuro

---

En este capítulo, se presentan las conclusiones finales de la investigación expuesta en la esta tesis doctoral. En estas conclusiones se incluyen las principales aportaciones realizadas en la misma, posteriormente se propone un conjunto de futuras líneas de investigación con las que se puede dar continuidad al trabajo realizado.

#### 7.1 Conclusiones

El proceso de describir una escena digital tal y como un humano la percibe no es una tarea fácil, debido a los múltiples factores que intervienen en el proceso: conocimiento previo, capacidad de análisis y reconocimiento. A pesar de que en la últimas décadas ha habido grandes avances al analizar una escena virtual reconociendo todos los objetos que aparecen en ella, esta tarea sigue siendo una de las más complejas de la visión por computador (Granholt, 2015; Márton and Türker, 2013; Prince, 2012; Szeliski, 2010). Este es precisamente el contexto en el que se desarrolla la presente tesis, para la cual se planteó un objetivo general (ver sección 1.3.1) y 6 específicos (ver sección 1.3.2). Como objetivo general se expuso la definición de un algoritmo de ingeniería inversa, capaz de reconocer elementos a partir de primitivas geométricas (específicamente cilindros, toros y esferas) en una nube de puntos pertenecientes a escenas escaneadas y asociarlos lógica, semántica y físicamente con elementos que se pueden encontrar en un entorno real. Dicho objetivo fue solventado con el análisis y generación del algoritmo propuesto, para el cual se desarrolló una investigación iniciando por el análisis del estado del arte sobre dos contextos, el primero se enfocó en el reconocimiento de objetos 3D, presentando el proceso general para el reconocimiento de objetos en nubes de puntos 3D, resaltando la importancia del uso

de descriptores visuales. Dicho proceso consta de seis subprocesos, dentro de los cuales se eligieron los cuatro más importantes (pre-proceso, extracción de descriptores, clasificación y validación) para el desarrollo de esta tesis. Además se detallaron los descriptores visuales existentes incluyendo ventajas, desventajas y comparativas, con el objetivo de estudiarlos y seleccionar el más adecuado para el algoritmo propuesto en esta tesis. Esta tarea fue importante ya que la selección de los descriptores visuales es uno de los pasos más críticos dentro del proceso de reconocimiento y depende de las características de los objetos que se pretende reconocer. Después del análisis se seleccionaron los descriptores visuales globales basados en voxel. Por otra parte se describieron los enfoques existentes de reconocimiento de objetos, enfoques basados en grafos topológicos y enfoque basado en geometría. Al final del análisis se concluyó en elegir el enfoque basado en geometría pero totalmente personalizado para que corresponda a los objetivos de la tesis.

La segunda sección del estado del arte correspondió a la presentación de las tecnologías semánticas, donde se estudió la evolución y actualidad de la Web semántica, tipos de ontologías y metodologías para su desarrollo y los lenguajes de implementación. El resultado del estudio de este contexto llevó a elegir a Methontology Gómez-Pérez (1998) como metodología para el desarrollo de la ontología propuesta. Finalmente se presentaron trabajos relacionados con la combinación de tecnologías semánticas y reconocimiento de formas. Lo cual, viendo las ventajas y carencias de los trabajos relacionados actuales, dio como resultado el planteamiento de las hipótesis 3.2 H1, H2 y H3. **H1:** *Es posible mejorar el reconocimiento de objetos a partir de la acotación semántica de primitivas que describan elementos comunes dentro de una escena industrial, específicamente cilindros, toros y esferas pertenecientes a una nube de puntos 3D considerando la combinación de descriptores de características geométricas y el uso de reglas semánticas.* **H2:** *Es posible enriquecer una nube de puntos detallando la definición de sus objetos de acuerdo al cálculo de sus características a partir de descriptores visuales.* **H3:** *La información proporcionada por este algoritmo puede aplicarse al reconocimiento de un entorno real.* Para lo cual se sugiere el desarrollo de un algoritmo que ayude a comprobar las hipótesis. Por lo que, el capítulo 4 presenta una descripción detallada del algoritmo propuesto el cual se compone de un modelo semántico y dos procesos: Reconocimiento de primitivas geométricas y Enriquecimiento semántico.

El modelo semántico se compone de una ontología y reglas semánticas que describirán el comportamiento de los objetos en una escena digital. El proceso de Reconocimiento de primitivas geométricas tendrá como entrada la nube de puntos y como salida una lista de segmentos clasificados de acuerdo a sus características geométricas. Mientras que el proceso de enriquecimiento semántico, obtendrá como resultado la descripción detallada de los elementos pertenecientes a la escena digital incluyendo su descripción lógica, física y semántica. **Información lógica** mediante el cálculo de características geométricas para cada objeto tales como altura, anchura, perímetro, radio, por mencionar algunas; mientras que la **información semántica** es obtenida haciendo coincidir cada objeto con

su elemento correspondiente en la ontología propuesta; finalmente la **información física** se obtiene mediante el cálculo de la topología de los elementos pertenecientes a la escena.

Una vez definido el algoritmo, se presentó en el capítulo 5 la validación del mismo, implementado cada uno de los procesos previamente definidos dentro de un contexto industrial y utilizando nubes de puntos de escenas industriales reales.

Los procesos del algoritmo fueron implementados y descritos detalladamente, en donde:

1. El proceso de reconocimiento de primitivas. Tiene el objetivo de responder a la pregunta *¿Existe algún objeto conocido dentro de la escena escaneada?*, para esto, se definieron tres subprocesos:
  - (a) Pre-proceso. Se encargó de eliminar el ruido de la nube de puntos industrial original
  - (b) Definición de descriptores. Se predefinieron descriptores visuales geométricos para el contexto industrial (Cilindro, esfera y toro).
  - (c) Clasificación. En este subproceso se realizó un estudio de tres algoritmos de reconocimiento de formas geométricas, con el fin de seleccionar uno para la implementación del algoritmo propuesto en la presente tesis. El algoritmo HFP fue el elegido de acuerdo a los resultados presentados en la Tabla 5.2.4, donde HFP mostró un comportamiento constante a lo largo de los casos de uso y todas las primitivas geométricas.

En este proceso se obtuvo como resultado una lista de segmentos clasificados (ver tabla 5.2.6) de acuerdo a su semejanza con las primitivas geométricas predefinidas con una etiqueta simple. Con la implementación de este proceso es posible responder la pregunta objetivo, por lo que, la respuesta es "Si, existen 57 objetos conocidos en la escena industrial, de los cuales 36 son Cilindros, 10 son Toros y 11 Esferas".

2. Proceso de enriquecimiento semántico. Este proceso tiene el objetivo de responder la pregunta *¿Qué tipo de elementos tiene la escena escaneada?*, para contestar dicha pregunta, es necesario implementar dos subprocesos y un modelo semántico:
  - (a) Modelo semántico, el cual está compuesto por una ontología personalizada de acuerdo al contexto y reglas semánticas. Dicha ontología fue desarrollada siguiendo la metodología llamada Methontology. Las reglas y la ontología fueron definidas de tal manera que representen el comportamiento real de los elementos de una escena industrial real. La ontología final se muestra en la figura 5.2.21.
  - (b) Cálculo de características geométricas, en este subproceso se definen y posteriormente calculan las propiedades que se calculan para cada figura geométrica. La tabla 5.2.13 muestra las propiedades geométricas que se calcularon de acuerdo con el tipo de primitiva geométrica.

- (c) Cálculo de la topología, en este subproceso se definen las relaciones de los elementos para generar una topología, considerando relaciones espaciales y representaciones espaciales de los elementos, estas relaciones son creadas a partir de la implementación de reglas semántica creadas para cada relación. La tabla 5.2.16 muestra las afirmaciones, *Spatial Relationship Rules*, *Spatial Representation Rules* y *Union Rules* que se aplicaron a cada segmento de la nube de puntos de prueba.

Por otra parte, se presentó una validación semántica del modelo semántico, en el cual se pudo observar tres diferentes casos que fueron resueltos con la información lógica, física y semántica calculada en la implementación del algoritmo.

Una vez implementado y validado el algoritmo, se presentó una evaluación cuantitativa, la cual tuvo el objetivo de medir la calidad de clasificación del algoritmo comparándolo con un algoritmo de reconocimiento de primitivas geométricas tradicional con el objetivo de comprobar las hipótesis H1, H2 y H3.

- **H1:** *Es posible **mejorar** el reconocimiento de objetos a partir de la **acotación semántica** de primitivas que describan elementos comunes dentro de una escena industrial, específicamente cilindros, toros y esferas pertenecientes a una nube de puntos 3D considerando la combinación de descriptores de características geométricas y el uso de reglas semánticas.* Por lo que, es comprobada a partir del desarrollo, implementación, verificación y evaluación del algoritmo en el contexto industrial. En la tabla 6.4.6 se puede observar que el algoritmo propuesto mejora los resultados de un algoritmo de reconocimiento de primitivas tradicional. Si se compara el espacio ROC del algoritmo tradicional (ver figura 6.4.1) y el espacio ROC del algoritmo propuesto (ver figura 6.4.2), también se puede observar que mejora la calidad de la segmentación al consideraran descriptores visuales de características geométricas y tecnología semántica.
- **H2:** *Es posible enriquecer una nube de puntos detallando la definición de sus objetos de acuerdo al cálculo de sus características a partir de descriptores visuales.* Esta hipótesis es comprobada a partir de los resultados obtenidos tanto en la validación como en la evaluación del algoritmo propuesto. Las figuras 5.3.1, 5.3.2 y 5.3.3 muestran cómo el algoritmo propuesto enriquece la maqueta digital al agregar información fehaciente tanto lógica, como física y semántica.
- **H3:** *La información proporcionada por este algoritmo puede aplicarse al reconocimiento de un entorno real.* Dicha hipótesis se comprueba a la largo de la implementación del algoritmo propuesto, donde se pudo observar el modelo en un contexto industrial real. La figura 5.2.21 presenta el grafo generado por la implementación del algoritmo donde se pueden observar las relaciones entre los elementos industriales reales con un comportamiento semejante a la realidad.

Con todo lo anterior en mente, es posible afirmar el cumplimiento de los objetivos general y específicos. El primero al describir, implementar y validar el algoritmo propuesto en esta tesis doctoral, el cual cumple con la generación, por cada escena escaneada, de una maqueta digital enriquecida en donde todos sus objetos fueron descritos con un alto grado de detalle, proporcionando su forma, sus características geométricas como alto, largo, diámetro; su topología representando las relaciones espaciales entre los objetos y describiendo semánticamente cada uno de acuerdo al rol que representa cada elemento en la escena. Mientras que, se dió cumplimiento a los objetivos específicos de acuerdo a:

- *El algoritmo debe estar conformado por módulos funcionales correctamente identificados y acotados.* EL algoritmo fue modulado en dos procesos generales (Reconocimiento de primitivas geométricas y Enriquesimiento semántico) con subprocesos funcionales cada uno (ver Figura 4.1.1).
- *Identificar y analizar el formato de la fuente de datos.* Se analizaron las fuentes de datos pertenecientes a escenas reales escaneadas. Específicamente los ficheros XYZ que contienen la lista de coordenadas de las nubes de puntos (ver sección ).
- *Incluir descriptores geométricos, los cuales representen y describan matemáticamente las primitivas a identificar.* Para la implementación del algoritmo en el contexto industrial se analizó descriptor visual del cilindro (ver sección 5.2.1.2). Tal como se comentó anteriormente, el cilindro es el elemento más frecuente en escenas industriales.
- *Diseñar y desarrollar el modelo semántico de objetos reales asociado a las primitivas encontradas en la nube de puntos, de manera que permitan asociar las entidades u objetos, con texto legible y entendido por humanos.* Este objetivo específico de cumplió al desarrollar la ontología y las reglas semánticas en la implementación del algoritmo (ver sección 5.2.2.1).
- *Definir reglas semánticas que permitan representar fiel y semánticamente a los elementos pertenecientes a una escena real.* Fueron definidas reglas semánticas que permitieron la realización de un clasificación correcta de los elementos. Estas reglas aseguran que el comportamiento de los elementos se asemeja a la realidad (ver sección 5.2.2.5).
- *Validar del algoritmo.* El algoritmo no solo fue validado a partir de la implementación del algoritmo propuesto en un entorno industrial real (ver el capítulo 5). Además se incluyó una evaluación cuantitativa del mismo (ver el capítulo 6).

Se creó un algoritmo de ingeniería inversa que procesa, reconoce y realiza anotaciones semánticas de elementos de una escena escaneada 3D, dicho algoritmo genera maquetas digitales enriquecidas con información lógica, física y semántica. El algoritmo presentado

permite una gran versatilidad de uso, es decir, proporciona la capacidad de adaptarse a diversos contextos con mínimas modificaciones, debido a que está basado en elementos compuestos de primitivas geométricas y éstos son muy comunes en el mundo real.

## 7.2 Líneas futuras de investigación

La investigación que conllevó el desarrollo de esta tesis ha propiciado una serie de posibles líneas futuras de investigación que permiten ampliar el trabajo realizado. Las primeras líneas futuras a considerar son en el ámbito del reconocimiento de primitivas geométricas, en donde es posible ampliar del tipo de formas simples a complejas, es decir, el reconocimiento de figuras compuestas predefinidas, para lo cual es necesario el desarrollo de los descriptores adecuados.

Dentro del ámbito del enriquecimiento semántico, sería interesante ampliar más el modelo semántico con reglas que describieran otro tipo de comportamiento en los objetos, por ejemplo número de elementos posibles dentro de toda la escena, significado de las relaciones entre estos elementos, es decir, que sea posible analizar una escena con el fin de evaluar los riesgos de seguridad potenciales que existan en ella, si existen objetos que estén obstruyendo puertas o ventanas o si existen objetos en posiciones inadecuadas.

Dentro del mismo ámbito, otra línea de investigación podría ser la de convertir la información almacenada en la ontología a Linked Data, para vincular los datos a fuentes externas de Linked Open Data, de esta manera tener una descripción semántica más amplia de los elementos.

Son diversas las líneas futuras de investigación que se pueden derivar a partir del trabajo de tesis presentado en este documento, en esta sección se han presentado las que se consideran, podrían complementar de manera relevante el trabajo de tesis doctoral presentado.

---

## Bibliografía

---

- Aldoma, A., M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, and G. Bradski (2011). “CAD-model recognition and 6DOF pose estimation using 3D cues”. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 585–592.
- As Ari, M., U. Sheikh, and E. Supriyanto (2014). “3D shape descriptor for object recognition based on Kinect-like depth image”. *Image and Vision Computing*.
- Attene, M., B. Falcidieno, and M. Spagnuolo (2006). “Hierarchical mesh segmentation based on fitting primitives”. *The Visual Computer*.
- Attene, M., S. Katz, M. Mortara, G. Patane, M. Spagnuolo, and A. Tal (2006). “Mesh Segmentation - A Comparative Study”. In: *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*. IEEE, pp. 7–7. DOI: 10.1109/SMI.2006.24.
- Attene, M. and F. Robbiano (2009). “Characterization of 3D shape parts for semantic annotation”. *Computer-Aided Design*.
- Attene, M. and G. Patane (2010). “Hierarchical Structure Recovery of Point-Sampled Surfaces”. *Computer Graphics Forum*, 29(6), pp. 1905–1920. DOI: 10.1111/j.1467-8659.2010.01658.x.
- Babinec, A., M. Dekan, F. Duchon, and A. Vitko (2012). “Modifications of VFH navigation methods for mobile robots”. *Procedia Engineering*.
- Bay, H., T. Tuytelaars, and L. V. Gool (2006). “Surf: Speeded up robust features”. *Computer vision ECCV 2006*.
- Berners-Lee (2006). *Artificial Intelligence and the Semantic Web*.
- Berners-Lee, T. (2001). “The semantic web”. *Scientific american*.

- Berners-Lee, T., J. Hendler, and O. Lassila (2001). “The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities”. *Scientific American*.
- Biasotti, S., S. Marini, and M. Mortara (2003a). “3D shape matching through topological structures”. *Discrete Geometry for Computer Imagery*, pp. 194–203.
- Biasotti, S., S. Marini, and M. Mortara (2003b). “An overview on properties and efficacy of topological skeletons in shape modelling”. *Shape Modeling and Applications, International Conference on*, p. 245.
- Biembengut, M. and N. Hein (1999). “Modelización matemática: estrategia para enseñar y aprender matemática”. *Educación Matemática* 11 (1).
- Bieniecki, W. (2007). “Image preprocessing for improving ocr accuracy”. *Perspective Technologies and Methods in MEMS Design, 2007. MEMSTECH 2007. International Conference on*, pp. 75–80.
- Blinn, J. (1978). “Simulation of wrinkled surfaces”. *ACM SIGGRAPH computer graphics*.
- Cabral, G., F. Pachet, J. Briot, and S. Paris (2005). “Automatic X traditional descriptor extraction: The case of chord recognition”. *Proceedings of the 6th international conference on music information retrieval (ISMIR’2005)*.
- Cajal, C., J. Santolaria, and J. Velazquez (2013). “Volumetric error compensation technique for 3D printers”. *Procedia Engineering*, 63, pp. 642–649.
- Chaperon, T., F. Goulette, and C. Laurgeau (2001). “Extracting Cylinders in Full 3D Data Using a Random Sampling Method and the Gaussian Image.” *VMV*.
- Chen, D., L. Zhang, P. T. Mathiopoulos, and X. Huang (2014). “A Methodology for Automated Segmentation and Reconstruction of Urban 3-D Buildings from ALS Point Clouds”. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(10), pp. 4199–4217. DOI: 10.1109/JSTARS.2014.2349003.
- Chen, X., A. Golovinskiy, and T. Funkhouser (2009). “A benchmark for 3D mesh segmentation”. *ACM Transactions on Graphics*, 28(3), p. 1. DOI: 10.1145/1531326.1531379.
- Chikofsky, E. and J. Cross (1990). “Reverse engineering and design recovery: A taxonomy”. *Software, IEEE*.
- Corcho, O., M. Fernández-López, and A. Gómez-Pérez (2003). “Methodologies, tools and languages for building ontologies. Where is their meeting point?” *Data & Knowledge Engineering*, 46(1), pp. 41–64. DOI: 10.1016/S0169-023X(02)00195-7.
- Cormen, T. (2009). “Introduction to algorithms”.
- Cui, W., J. Kannan, and H. Wang (2007). “Discoverer: Automatic Protocol Reverse Engineering from Network Traces.” *USENIX Security*.



- Dey, T., J. Giesen, and S. Goswami (2003). “Shape segmentation and matching with flow discretization”. *Algorithms and Data Structures*.
- Díaz, R., L. Gil, C. Serrano, and M. Blasco (2004). “Comparison of three algorithms in the classification of table olives by means of computer vision”. *Journal of Food Engineering*, 61(1), pp. 101–107.
- Díaz-Vilariño, L., S. Lagüela, J. Armesto, and P. Arias (2013). “Semantic as-built 3d models including shades for the evaluation of solar influence on buildings”. *Solar Energy*, 92, pp. 269–279. DOI: 10.1016/j.solener.2013.03.017.
- Divvala, S., D. Hoiem, and J. Hays (2009). “An empirical study of context in object detection”. *Computer Vision and Pattern Recognition*, pp. 1271–1278.
- Dogmus, Z., E. Erdem, and V. Patoglu (2015). “RehabRobo-Onto: Design, development and maintenance of a rehabilitation robotics ontology on the cloud”. *Robotics and Computer-Integrated Manufacturing*, 33, pp. 100–109. DOI: 10.1016/j.rcim.2014.08.010.
- Dornbusch, T., P. Wernecke, and W. Diepenbrock (2007). “A method to extract morphological traits of plant organs from 3D point clouds as a database for an architectural plant model”. *Ecological Modelling*.
- Egenhofer, M. and R. Franzosa (1991). “Point-set topological spatial relations”. *International Journal of Geographical Information System*, 5(2), pp. 161–174.
- Egenhofer, M. and J. Herring (1990). “A mathematical framework for the definition of topological relationships”. *Fourth international symposium on spatial data handling*, pp. 803–813.
- Egenhofer, M. and J. Sharma (1993). “Topological relations between regions in  $p_2$  and  $z_2$ ”. *Advances in spatial databases*.
- Eilam, E. (2011). *Reversing: secrets of reverse engineering*.
- Erdos, G., T. Nakano, and J. Váncza (2014). “Adapting CAD models of complex engineering objects to measured point cloud data”. *CIRP Annals-Manufacturing Technology*.
- Fechteler, P., P. Eisert, and J. Rurainsky (2007). “Fast and high resolution 3d face scanning”. *Image Processing, 2007. ICIIP 2007. IEEE International Conference*, 3, pp. III–81.
- Feng, J., Y. Liu, and L. Gong (2015). “Junction aware shape descriptor for 3D articulated models using local shape radius variation”. *Signal Processing*.
- Fensel, D. (2001). “Ontologies”. *Springer Berlin Heidelberg*, pp. 11–18.
- Fiorini, S. R., J. L. Carbonera, P. Gonçalves, V. A. Jorge, V. F. Rey, T. Haidegger, M. Abel, S. A. Redfield, S. Balakirsky, V. Ragavan, H. Li, C. Schlenoff, and E. Prestes

- (2015). “Extensions to the core ontology for robotics and automation”. *Robotics and Computer-Integrated Manufacturing*, 33, pp. 3–11. DOI: 10.1016/j.rcim.2014.08.004.
- Gaither, N. and G. Frazier (2000). “Administración de producción y operaciones”.
- Gao, Y., J. Tang, H. Li, Q. Dai, and N. Zhang (2010). “View-based 3D model retrieval with probabilistic graph model”. *Neurocomputing*.
- Genesereth, M. and R. Fikes (1992). “Knowledge interchange format-version 3.0: reference manual”.
- Gennari, J. H., M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu (2003). “The evolution of Protege: an environment for knowledge-based systems development”. *International Journal of Human-Computer Studies*, 58(1), pp. 89–123. DOI: 10.1016/S1071-5819(02)00127-1.
- Golovinskiy, A. and T. Funkhouser (2008). “Randomized cuts for 3D mesh analysis”. In: *ACM SIGGRAPH Asia 2008 papers on - SIGGRAPH Asia '08*. Vol. 27. 5. New York, New York, USA: ACM Press, p. 1. DOI: 10.1145/1457515.1409098.
- Gómez-Pérez, A. (1998). “Knowledge sharing and reuse”. *Handbook of applied expert systems*.
- Gómez-Pérez, A., M. Fernández-López, and O. Corcho (2006). *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*.
- Gómez-Romero, J., M. A. Patricio, J. García, and J. M. Molina (2011). “Ontology-based context representation and reasoning for object tracking and scene interpretation in video”. *Expert Systems with Applications*, 38(6), pp. 7494–7510. DOI: 10.1016/j.eswa.2010.12.118.
- Gonzalez, R. (2009). “Digital image processing”.
- Gonzalez, R., R. Woods, and S. Eddins (2004). “Digital image processing using MATLAB”.
- Granholm, E. (2015). “Feature-based Image Interpretation for Lightweight Computer Vision”.
- Gruber, T. R. (1993). “A translation approach to portable ontology specifications”. *Knowledge Acquisition*, 5(2), pp. 199–220. DOI: 10.1006/knac.1993.1008.
- Grüninger, M. and M. Fox (1995). “The role of competency questions in enterprise engineering”. *Benchmarking-Theory and Practice*.
- Guarino, N. (1998). “Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy”.

- Guptill, S. and J. Morrison (2013). *Elements of spatial data quality*. Ed. by Elsevier.
- Haala, N. and M. Kada (2010). “An update on automatic 3D building reconstruction”. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6), pp. 570–580. DOI: 10.1016/j.isprsjprs.2010.09.006.
- Hakala, D., R. Hillyard, P. Malraison, and B. Nource (1981). “Natural quadrics in mechanical design”. *Proc-AUTOFACT West*.
- Heijst, G. V., A. Schreiber, and B. Wielinga (1997). “Using explicit ontologies in KBS development”. *International journal of human-computer studies*, 42(1), pp. 183–292.
- Hilaga, M. and Y. Shinagawa (2001). “Topology matching for fully automatic similarity estimation of 3D shapes”. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM*, pp. 203–212.
- Hmida, H. B., F. Boochs, F. Mainz, C. Cruz, and C. Nicolle (2012). “Knowledge Base Approach for 3D Objects Detection in Point Clouds Using 3D Processing and Specialists Knowledge”. 5(1), pp. 1–14.
- Hois, J., M. Wüstel, J. Bateman, and T. Röfer (2006). “Dialog-based 3D-image recognition using a domain ontology”. *Spatial Cognition V Reasoning, Action, Interaction. Springer Berlin Heidelberg*, pp. 107–126.
- Hong, C., J. Yu, J. You, X. Chen, and D. Tao (2015). “Multi-view ensemble manifold regularization for 3D object recognition”. *Information Sciences*, 320, pp. 395–405. DOI: 10.1016/j.ins.2015.03.032.
- Hong, T., S. D’Oca, W. J. Turner, and S. C. Taylor-Lange (2015). “An ontology to represent energy-related occupant behavior in buildings Part I: Introduction to the DNAs Framework”. *Building and Environment*, 92, pp. 764–777. DOI: 10.1016/j.buildenv.2015.02.019.
- Hoppe, H., T. DeRose, and T. Duchamp (1992). *Surface reconstruction from unorganized points*. Ed. by ACM.
- Hou, S., H. Li, and Y. Rezgui (2015). “Ontology-based approach for structural design considering low embodied energy and carbon”. *Energy and Buildings*, 102, pp. 75–90. DOI: 10.1016/j.enbuild.2015.04.051.
- Ivorra, E., A. Sánchez, and J. Camarasa (2015). “Assessment of grape cluster yield components based on 3D descriptors using stereo vision”. *Food Control*.
- Jensfelt, P. and D. Kragic (2006). “A framework for vision based bearing only 3D SLAM”. *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on. IEEE*, pp. 1944–1950.

- Jia, X. and J. Richards (1999). “Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification”. *Geoscience and Remote Sensing, IEEE Transactions on*, 37(1), pp. 538–542.
- Kang, T., I. Choi, and M. Lim (2015). “MDGHM-SURF: A robust local image descriptor based on modified discrete Gaussian Hermite moment”. *Pattern Recognition*.
- Kaufman, A. and E. Shimony (1987). “3D scan conversion algorithms for voxel based graphics”. *Proceedings of the 1986 workshop on Interactive 3D graphics*. ACM, pp. 45–75.
- Kobbelt, J. W. L. (2005). “Structure recovery via hybrid variational surface approximation”. *Computer Graphics Forum*.
- Kumagai, J. (2000). “Chip detectives [reverse engineering]”. *IEEE Spectrum*, 37(11), pp. 43–48. DOI: 10.1109/6.880953.
- Lam, M. (2005). “Seminars in Child and Adolescent Psychiatry”. *Hong Kong Journal of Psychiatry*.
- Lee, D. and K. Sabnani (1993). “Reverse-engineering of communication protocols”. *Network Protocols, 1993. Proceedings., 1993 International Conference on*. IEEE, pp. 208–216.
- Lee, K. C., Y. Bae, and S. Lee (2015). “Integration of ontology-based decision support and a cognitive map approach to predict flat-screen TV market trends”. *Technological Forecasting and Social Change*, 95, pp. 109–119. DOI: 10.1016/j.techfore.2015.02.013.
- Leifman, G., R. Meir, and A. Tal (2005). “Semantic-oriented 3d shape retrieval using relevance feedback”. *The Visual Computer*.
- Lenat, D. and R. Guha (1993). “Building large knowledge-based systems: Representation and inference in the CYC project”. *Artificial Intelligence*.
- Léon, V., N. Bonneel, G. Lavoué, and J.-P. Vandeboire (2016). “Continuous semantic description of 3D meshes”. *Computers & Graphics*, 54, pp. 47–56. DOI: 10.1016/j.cag.2015.07.018.
- Levine, D. M., N. K. Dutta, J. Eckels, C. Scanga, C. Stein, S. Mehra, D. Kaushal, P. C. Karakousis, and H. Salamon (2015). “A tuberculosis ontology for host systems biology”. *Tuberculosis*. DOI: 10.1016/j.tube.2015.05.012.
- Levinson, S. (1996). “Frames of reference and Molyneux’s question: Crosslinguistic evidence”. *Language and space*.
- Li, C., J. Lu, C. Yin, and L. Ma (2009). “Qualitative Spatial Representation and Reasoning in 3D Space”. In: *2009 Second International Conference on Intelligent Computation*

- Technology and Automation*. Vol. 1. IEEE, pp. 653–657. DOI: 10.1109/ICICTA.2009.163.
- Li, H., Z. Liu, Y. Huang, and Y. Shi (2015). “Quaternion generic Fourier descriptor for color object recognition”. *Pattern Recognition*, 48(12), pp. 3895–3903. DOI: 10.1016/j.patcog.2015.06.002.
- Li, Y., W. Liu, X. Li, and Q. Huang (2014). “GA-SIFT: A new scale invariant feature transform for multispectral image using geometric algebra”. *Information Sciences*.
- Li, Y., X. Wu, Y. Chrysathou, and A. Sharf (2011). “Globfit: Consistently fitting primitives by discovering global relations”. *ACM Transactions on Graphics (TOG)*, 30(4), p. 52.
- Liao, K., G. Liu, and Y. Hui (2013). “An improvement to the SIFT descriptor for image representation and matching”. *Pattern Recognition Letters*, 34(11), pp. 1211–1220. DOI: 10.1016/j.patrec.2013.03.021.
- Lin, C., J. Chen, P. Su, and C. Chen (2014). “Eigen-feature analysis of weighted covariance matrices for LiDAR point cloud classification”. *ISPRS Journal of Photogrammetry and Remote Sensing*, 94, pp. 70–79.
- Liu, L. and M. Zsu (2009). *Encyclopedia of database systems*.
- López, G., V. Custodio, J. Moreno, M. Sikora, P. Moura, and N. Fernández (2015). “Modeling Smart Grid neighborhoods with the ENERSip ontology”. *Computers in Industry*, 70, pp. 168–182. DOI: 10.1016/j.compind.2015.01.008.
- Lowe, D. (1999). “Object recognition from local scale-invariant features”. *the proceedings of the seventh IEEE international conference on*, 2, pp. 1150–1157.
- Lowe, D. (2004). “Distinctive image features from scale-invariant keypoints”. *International journal of computer vision*, 60(2), pp. 91–110.
- Mair, E., G. Hager, and D. Burschka (2010). “Adaptive and generic corner detection based on the accelerated segment test”. *Computer Vision–ECCV 2010. Springer Berlin Heidelberg*, pp. 183–196.
- Marshall, G. and G. Stutz (2011). *Handbook of optical and laser scanning*.
- Marton, Z. and D. Pangercic (2010). “Hierarchical object geometric categorization and appearance classification for mobile manipulation”. *International Conference on Humanoid Robots (Humanoids), 2010 10th IEEE-RAS*.
- Márton, Z. and S. Türker (2013). “On Bayesian Inference for Embodied Perception of Object Poses”. *Intelligent Robots and Systems, 2013 IEEE/RSJ International Conference on, Workshop on Metrics of Embodied Learning Processes in Robots and Animals*.

- Masood, S. and W. Rattanawong (2002). “A generic part orientation system based on volumetric error in rapid prototyping”. *The International Journal of Advanced Manufacturing Technology*, 19(3), pp. 209–216.
- Masood, S., W. Rattanawong, and P. Iovenitti (2000). “Part build orientations based on volumetric error in fused deposition modelling”. *The International Journal of Advanced Manufacturing Technology*, 16(3), pp. 162–168.
- Matzko, S., P. Clarke, and T. Gibbs (2002). “Reveal: A tool to reverse engineer class diagrams”. *Proceedings of the Fortieth International Conference on Tools Pacific: Objects for internet, mobile and embedded applications*. Australian Computer Society, Inc, pp. 13–21.
- McAdams, A., A. Selle, R. Tamstorf, and J. Teran (2011). “Computing the Singular Value Decomposition of 3X 3 matrices with minimal branching and elementary floating point operations”.
- Méndez, V., J. R. Rosell-Polo, R. Sanz, A. Escolà, and H. Catalán (2014). “Deciduous tree reconstruction algorithm based on cylinder fitting from mobile terrestrial laser scanned point clouds”. *Biosystems Engineering*, 124, pp. 78–88. DOI: 10.1016/j.biosystemseng.2014.06.001.
- Moore, A. and T. Hall (1990). “Efficient memory-based learning for robot control”.
- Moratz, R. and M. Ragni (2008). “Qualitative spatial reasoning about relative point position”. *Journal of Visual Languages and Computing*, 19(1), pp. 75–98. DOI: 10.1016/j.jvlc.2006.11.001.
- Moratz, R., T. Tenbrink, J. Bateman, and K. Fischer (2003). “Spatial knowledge representation for human-robot interaction”. *Spatial cognition III*, pp. 1035–1035.
- Moret, B. and H. Shapiro (2001). “Algorithms and experiments: The new (and old) methodology”. *Journal of Universal Computer Science*.
- Mortara, M. and G. Patané (2004). “Plumber: a method for a multi-scale decomposition of 3D shapes into tubular primitives and bodies”. *Proceedings of the ninth ACM symposium on Solid modeling and applications*, pp. 339–344.
- Mortara, M., G. Patané, and M. Spagnuolo (2006). “From geometric to semantic human body models”. *Computers and Graphics (Pergamon)*, 30(2), pp. 185–196. DOI: 10.1016/j.cag.2006.01.024.
- Mountrakis, G., J. Im, and C. Ogole (2011). “Support vector machines in remote sensing: A review”. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(3), pp. 247–259.
- Mugzach, O., M. Peleg, S. C. Bagley, S. J. Guter, E. H. Cook, and R. B. Altman (2015). “An ontology for Autism Spectrum Disorder (ASD) to infer ASD phenotypes from

- Autism Diagnostic Interview-Revised data.” *Journal of biomedical informatics*, 56, pp. 333–347. DOI: 10.1016/j.jbi.2015.06.026.
- Nan, L., A. Sharf, and H. Zhang (2010). “SmartBoxes for interactive urban reconstruction”. *ACM SIGGRAPH 2010 papers on - SIGGRAPH ’10*, 29(4), p. 1.
- Nohl, K., D. Evans, S. Starbug, and H. Plötz (2008). “Reverse-Engineering a Cryptographic RFID Tag.” *USENIX security symposium*.
- Noy, N. F., R. W. Fergerson, and M. A. Musen (2000). “The Knowledge Model of Protege-2000: Combining Interoperability and Flexibility”, pp. 17–32.
- Noy, N. and D. McGuinness (2001). “Ontology development 101: A guide to creating your first ontology”.
- Nuttens, T., A. D. Wulf, and G. Deruyter (2012). “Application of laser scanning for deformation measurements: a comparison between different types of scanning instruments.” *Proceedings of the FIG Working Week*.
- Onuh, S., N. Bennett, and V. Hughes (2006). “Reverse engineering and rapid tooling as enablers of agile manufacturing”. *International Journal of Agile Systems and Management*, 1(1), pp. 60–72.
- Otto, K. and K. Wood (1998). “Product evolution: a reverse engineering and redesign methodology”. *Research in Engineering Design*.
- Pauly, M., M. Gross, and L. Kobbelt (2002). “Efficient simplification of point-sampled surfaces”. . . . *of the conference on Visualization’02*.
- Petitjean, S. (2002). “A survey of methods for recovering quadrics in triangle meshes”. *ACM Computing Surveys*, 34(2), pp. 211–262. DOI: 10.1145/508352.508354.
- Ponce, J., M. Hebert, C. Schmid, and A. Zisserman (2007). *Toward category - level object recognition*.
- Prince, S. (2012). *Computer vision: models, learning, and inference*. Ed. by Cambridge University Press.
- Rabbani, T. and F. V. D. Heuvel (2005). “Efficient hough transform for automatic detection of cylinders in point clouds”. *ISPRS WG III/3, III/4*.
- Reed, S. and D. Lenat (2002). “Mapping ontologies into Cyc”. *AAAI 2002 Conference Workshop on Ontologies For The Semantic Web*.
- Requicha, A. and H. Voelcker (1982). “Solid modeling: a historical summary and contemporary assessment”. *IEEE computer graphics and applications*.
- Rosten, E. and T. Drummond (2006). “Machine learning for high - speed corner detection”. *Computer Vision ECCV 2006*.

- Rubio, J. C., A. Eigenstetter, and B. Ommer (2015). “Generative regularization with latent topics for discriminative object recognition”. *Pattern Recognition*, 48(12), pp. 3871–3880. DOI: 10.1016/j.patcog.2015.06.013.
- Ruiz-Sarmiento, J.-R., C. Galindo, and J. Gonzalez-Jimenez (2015). “Scene object recognition for mobile robots through Semantic Knowledge and Probabilistic Graphical Models”. *Expert Systems with Applications*, 42(22), pp. 8805–8816. DOI: 10.1016/j.eswa.2015.07.033.
- Rusu, R. B., N. Blodow, Z. C. Marton, and M. Beetz (2009). “Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation in domestic environments”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 1–6. DOI: 10.1109/IRROS.2009.5354683.
- Rusu, R. (2010). “Semantic 3d object maps for everyday manipulation in human living environments”. *KI-Kunstliche Intelligenz*.
- Rusu, R. and Z. Marton (2009). “Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments”. *Intelligent Robots and Systems*.
- Sankaranarayanan, J., H. Samet, and A. Varshney (2007). “A fast all nearest neighbor algorithm for applications involving large point-clouds”. *Computers & Graphics*.
- Schnabel, R., R. Wahl, and R. Klein (2007). “Efficient RANSAC for Point-Cloud Shape Detection”. *Computer Graphics Forum*, 26(2), pp. 214–226. DOI: 10.1111/j.1467-8659.2007.01016.x.
- Sengupta, S., E. Greveson, A. Shahrokni, and P. H. S. Torr (2013). “Urban 3D semantic modelling using stereo vision”. *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 580–585. DOI: 10.1109/ICRA.2013.6630632.
- Shadbolt, N., W. Hall, and T. Berners-Lee (2006). “The semantic web revisited”. *Intelligent Systems, IEEE*.
- Shamir, A. (2006). “Segmentation and shape extraction of 3D boundary meshes”. *State-of-the-Art Report, Proceedings Eurographics*.
- Shevertalov, M. and S. Mancoridis (2007). “A reverse engineering tool for extracting protocols of networked applications”. *Reverse Engineering, 2007. WCRE 2007. 14th Working Conference on. IEEE*, pp. 229–238.
- Sitek, A., R. Huesman, and G. Gullberg (2006). “Tomographic reconstruction using an adaptive tetrahedral mesh defined by a point cloud”. *IEEE Transactions on Medical Imaging*, 25(9), pp. 1172–1179. DOI: 10.1109/TMI.2006.879319.
- Steinbach, E. and B. Girod (2000). “3D object reconstruction using spatially extended voxels and multi-hypothesis voxel coloring”. *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, 1, pp. 774–777.



- Steiner, D. and A. Fischer (2002). “Cutting 3D freeform objects with genus- $n$  into single boundary surfaces using topological graphs”. *Proceedings of the seventh ACM symposium on Solid modeling and applications*. ACM, pp. 336–343.
- Suárez-Figueroa, M. (2009). “How to write and use the Ontology Requirements Specification Document”. *On the move to meaningful internet systems: OTM 2009*. Springer Berlin Heidelberg, pp. 966–982.
- Sung, K. and T. Poggio (1998). “Example-based learning for view-based human face detection”. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(1), pp. 39–51.
- Sykora, P., P. Kamencay, and R. Hudec (2014). “Comparison of SIFT and SURF Methods for Use on Hand Gesture Recognition based on Depth Map”. *AASRI Procedia*, 9, pp. 19–24. DOI: 10.1016/j.aasri.2014.09.005.
- Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.
- Tamayo, M. (2004). *El proceso de la investigación científica*. Ed. by Limusa.
- Tangelder, J. and R. Veltkamp (2008). “A survey of content based 3D shape retrieval methods”. *Multimedia tools and applications*.
- Tombari, F., S. Salti, and L. Stefano (2011). “A combined texture-shape descriptor for enhanced 3D feature matching”. *Image Processing (ICIP), 2011 18th IEEE International Conference on. IEEE*, pp. 809–812.
- Tran, T.-T., V.-T. Cao, and D. Laurendeau (2015). “Extraction of cylinders and estimation of their parameters from point clouds”. *Computers & Graphics*, 46, pp. 345–357. DOI: 10.1016/j.cag.2014.09.027.
- Uschold, M. and M. King (1995). *Towards a methodology for building ontologies*. Ed. by U. o. E. Edinburgh: Artificial Intelligence Applications Institute.
- Varady, T., R. Martin, and J. Cox (1997). “Reverse engineering of geometric models-an introduction”. *Computer-Aided Design*.
- Vosselman, G. and B. Gorte (2004). “Recognising structure in laser scanner point clouds”. *International archives of photogrammetry, remote sensing and spatial information sciences*, 46(8), pp. 33–38.
- Wohlkinger, W. and M. Vincze (2011). “Shape distributions on voxel surfaces for 3D object classification from depth images”. *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*.

- Woo, H., E. Kang, S. Wang, and K. H. Lee (2002). “A new segmentation method for point cloud data”. *International Journal of Machine Tools and Manufacture*, 42(2), pp. 167–178. DOI: 10.1016/S0890-6955(01)00120-1.
- Wu, J. and J. Rehg (2011). “CENTRIST: A visual descriptor for scene categorization”. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8), pp. 1489–1501.
- Wu, X., Q. Zhao, and W. Bu (2014). “A SIFT-based contactless palmprint verification approach using iterative RANSAC and local palmprint descriptors”. *Pattern Recognition*.
- Wu, Y., Y. Liu, Z. Yuan, and N. Zheng (2012). “IAIR-CarPed: A psychophysically annotated dataset with fine-grained and layered semantic labels for object recognition”. *Pattern Recognition Letters*, 33(2), pp. 218–226. DOI: 10.1016/j.patrec.2011.10.003.
- Yang, Y., H. Lin, and Q. Zhu (2004). “Content-based 3D model retrieval: a survey”. *CHINESE JOURNAL OF COMPUTERS-CHINESE EDITION*, 27(10), pp. 1297–1310.
- Yao, A. (2005). “Applications of 3D scanning and reverse engineering techniques for quality control of quick response products”. *The International Journal of Advanced Manufacturing Technology*, 26(11-12), pp. 1284–1288.
- Zlatanova, S., A. A. Rahman, and W. Shi (2004). “Topological models and frameworks for 3D spatial objects”. *Computers and Geosciences*, 30(4), pp. 419–428. DOI: 10.1016/j.cageo.2003.06.004.
- Zuckerberger, E., A. Tal, and S. Shlafman (2002). “Polyhedral surface decomposition with applications”. *Computers & Graphics*.